1

Introduction: Human and Computational Mind

In this Chapter we compare and contrast human and computational mind, from psychological, AI and CI perspectives.

1.1 Natural Intelligence and Human Mind

Recall that the word *intelligence* (plural *intelligences*) comes from Latin *intellegentia*.[1] It is a property of *human mind* that encompasses many related *mental abilities*, such as the capacities to *reason*, *plan*, solve problems, think abstractly, comprehend ideas and *language*, and learn. Although many regard the concept of intelligence as having a much broader scope, for example in *cognitive science* and *computer science*, in some schools of *psychology*,[2] the

[1] *Intellegentia* is a combination of Latin *inter* = *between* and *legere* = *choose, pick out, read.* Inter–lege–nt–ia, literally means 'choosing between.'

Also, note that there is a is a scientific journal titled 'Intelligence', dealing with intelligence and psychometrics. It was founded in 1977 by Douglas K. Detterman of Case Western Reserve University. It is currently published by Elsevier and is the official journal of the International Society for Intelligence Research.

[2] Recall that *psychology* is an academic and applied field involving the study of the human mind, brain, and behavior. Psychology also refers to the application of such knowledge to various spheres of human activity, including problems of individuals' daily lives and the treatment of mental illness.

Psychology differs from anthropology, economics, political science, and sociology in seeking to explain the mental processes and behavior of individuals. Psychology differs from biology and neuroscience in that it is primarily concerned with the interaction of mental processes and behavior, and of the overall processes of a system, and not simply the biological or neural processes themselves, though the subfield of neuropsychology combines the study of the actual neural processes with the study of the mental effects they have subjectively produced.

The word psychology comes from the ancient Greek 'psyche', which means 'soul' or 'mind' and 'ology', which means 'study'.

V.G. Ivancevic and T.T. Ivancevic: *Introduction: Human and Computational Mind*, Studies in Computational Intelligence (SCI) **60**, 1–269 (2007)
www.springerlink.com

study of intelligence generally regards this trait as distinct from *creativity*, *personality*, *character*, or *wisdom*.

Briefly, the word *intelligence* has five common meanings:

1. Capacity of human mind, especially to understand principles, truths, concepts, facts or meanings, acquire knowledge, and apply it to practise; the ability to learn and comprehend.
2. A form of life that has such capacities.
3. Information, usually secret, about the enemy or about hostile activities.
4. A political or military department, agency or unit designed to gather such information.
5. Biological intelligent behavior represents animal's ability to make productive decisions for a specific task, given a root objective; this decision is based on learning which requires the ability to hold onto results from previous tasks, as well as being able to analyze the situation; the root objective for living organisms is simply survival; the 'specific task' could be a choice of food, i.e., one that provides long steady supply of energy as it could be a long while before the next mealtime; this is in perfect harmony with the root biological objective – survival.

According to Encyclopedia Britannica, *intelligence* is the *ability to adapt effectively to the environment, either by making a change in oneself or by changing the environment or finding a new one.* Different investigators have emphasized different aspects of intelligence in their definitions. For example, in a 1921 symposium on the definition of intelligence, the American psychologist Lewis Terman emphasized the *ability to think abstractly*, while another American psychologist, Edward Thorndike, emphasized *learning* and the ability to give good responses to questions. In a similar 1986 symposium, however, psychologists generally agreed on the importance of adaptation to the environment as the key to understanding both what intelligence is and what it does. Such adaptation may occur in a variety of environmental situations. For example, a student in school learns the material that is required to pass or do well in a course; a physician treating a patient with an unfamiliar disease adapts by learning about the disease; an artist reworks a painting in order to make it convey a more harmonious impression. For the most part, adapting involves making a change in oneself in order to cope more effectively, but sometimes effective adaptation involves either changing the environment or finding a new environment altogether. Effective adaptation draws upon a number of cognitive processes, such as perception, learning, memory, reasoning, and problem solving. The main trend in defining intelligence, then, is that it is not itself a cognitive or mental process, but rather a selective combination of these processes purposively directed toward effective adaptation to the environment. For example, the physician noted above learning about a new disease adapts by perceiving material on the disease in medical literature, learning what the material contains, remembering crucial aspects of it that are needed to treat the patient, and then reasoning to solve the problem of how

to apply the information to the needs of the patient. Intelligence, in sum, has come to be regarded as not a single ability but an effective drawing together of many abilities. This has not always been obvious to investigators of the subject, however, and, indeed, much of the history of the field revolves around arguments regarding the nature and abilities that constitute intelligence.

Now, let us quickly reflect on the above general *intelligence–related keywords*.

Reason

Recall that in the philosophy of arguments, *reason* is the ability of the human mind to form and operate on concepts in abstraction, in varied accordance with rationality and logic —terms with which reason shares heritage. Reason is thus a very important word in Western intellectual history, to describe a type or aspect of mental thought which has traditionally been claimed as distinctly human, and not to be found elsewhere in the animal world. Discussion and debate about the nature, limits and causes of reason could almost be said to define the main lines of historical philosophical discussion and debate. Discussion about reason especially concerns:

(a) its relationship to several other related concepts: language, logic, consciousness etc,
(b) its ability to help people decide what is true, and
(c) its origin.

The concept of reason is connected to the concept of language, as reflected in the meanings of the Greek word 'logos', later to be translated by Latin 'ratio' and then French 'raison', from which the English word derived. As reason, rationality, and logic are all associated with the ability of the human mind to predict effects as based upon presumed causes, the word 'reason' also denotes a ground or basis for a particular argument, and hence is used synonymously with the word 'cause'.

It is sometimes said that the contrast between reason and logic extends back to the time of Plato[3] and Aristotle[4]. Indeed, although they had no

[3] Plato (c. 427 — c. 347 BC) was an immensely influential ancient Greek philosopher, a student of Socrates, writer of philosophical dialogues, and founder of the Academy in Athens where Aristotle studied. Plato lectured extensively at the Academy, and wrote on many philosophical issues, dealing especially in politics, ethics, metaphysics, and epistemology. The most important writings of Plato are his dialogues, although some letters have come down to us under his name. It is believed that all of Plato's authentic dialogues survive. However, some dialogues ascribed to Plato by the Greeks are now considered by the consensus of scholars to be either suspect (e.g., First Alcibiades, Clitophon) or probably spurious (such as Demodocus, or the Second Alcibiades). The letters are all considered to probably be spurious, with the possible exception of the Seventh Letter. Socrates is often a character in Plato's dialogues. How much of the content and argument of any given dialogue is Socrates' point of view, and how much of it is Plato's, is

separate Greek word for logic as opposed to language and reason, Aristotle's *syllogism* (Greek 'syllogismos') identified logic clearly for the first time as a distinct field of study: the most peculiarly reasonable ('logikê') part of reasoning, so to speak.

---

heavily disputed, since Socrates himself did not write anything; this is often referred to as the 'Socratic problem'. However, Plato was doubtless strongly influenced by Socrates' teachings.

Platonism has traditionally been interpreted as a form of metaphysical dualism, sometimes referred to as Platonic realism, and is regarded as one of the earlier representatives of metaphysical objective idealism. According to this reading, Plato's metaphysics divides the world into two distinct aspects: the *intelligible world* of 'forms', and the *perceptual world* we see around us. The perceptual world consists of imperfect copies of the intelligible forms or ideas. These forms are unchangeable and perfect, and are only comprehensible by the use of the intellect or understanding, that is, a capacity of the mind that does not include sense-perception or imagination. This division can also be found in Zoroastrian philosophy, in which the dichotomy is referenced as the *Minu* (intelligence) and *Giti* (perceptual) worlds. Currently, in the domain of mathematical physics, this view has been adopted by Sir Roger Penrose [Pen89].

[4] Aristotle (384 BC — March 7, 322 BC) was an ancient Greek philosopher, a student of Plato and teacher of Alexander the Great. He wrote books on divers subjects, including physics, poetry, zoology, logic, rhetoric, government, and biology, none of which survive in their entirety. Aristotle, along with Plato and Socrates, is generally considered one of the most influential of ancient Greek philosophers. They transformed Presocratic Greek philosophy into the foundations of Western philosophy as we know it. The writings of Plato and Aristotle founded two of the most important schools of Ancient philosophy.

Aristotle valued knowledge gained from the senses and in modern terms would be classed among the modern empiricists. He also achieved a 'grounding' of dialectic in the Topics by allowing interlocutors to begin from commonly held beliefs (Endoxa), with his frequent aim being to progress from 'what is known to us' towards 'what is known in itself' (Physics). He set the stage for what would eventually develop into the empirical scientific method some two millennia later. Although he wrote dialogues early in his career, no more than fragments of these have survived. The works of Aristotle that still exist today are in treatise form and were, for the most part, unpublished texts. These were probably lecture notes or texts used by his students, and were almost certainly revised repeatedly over the course of years. As a result, these works tend to be eclectic, dense and difficult to read. Among the most important ones are Physics, Metaphysics (or Ontology), Nicomachean Ethics, Politics, De Anima (On the Soul) and Poetics. These works, although connected in many fundamental ways, are very different in both style and substance.

Aristotle is known for being one of the few figures in history who studied almost every subject possible at the time, probably being one of the first polymaths. In science, Aristotle studied anatomy, astronomy, economics, embryology, geography, geology, meteorology, physics, and zoology. In philosophy, Aristotle

No philosopher of any note has ever argued that logic is the same as reason. They are generally thought to be distinct, although logic is one important aspect of reason. But the tendency to the preference for 'hard logic', or 'solid logic', in modern times has incorrectly led to the two terms occasionally being

---

wrote on aesthetics, ethics, government, metaphysics, politics, psychology, rhetoric and theology. He also dealt with education, foreign customs, literature and poetry. His combined works practically constitute an encyclopedia of Greek knowledge. According to Aristotle, everything is made out of the five basic elements:

1. Earth, which is cold and dry;
2. Water, which is cold and wet;
3. Fire, which is hot and dry;
4. Air, which is hot and wet; and
5. Aether, which is the divine substance that makes up the heavenly spheres and heavenly bodies (stars and planets).

Aristotle defines his philosophy in terms of essence, saying that philosophy is 'the science of the universal essence of that which is actual'. Plato had defined it as the 'science of the idea', meaning by idea what we should call the unconditional basis of phenomena. Both pupil and master regard philosophy as concerned with the universal; Aristotle, however, finds the universal in particular things, and called it the essence of things, while Plato finds that the universal exists apart from particular things, and is related to them as their prototype or exemplar. For Aristotle, therefore, philosophic method implies the ascent from the study of particular phenomena to the knowledge of essences, while for Plato philosophic method means the descent from a knowledge of universal ideas to a contemplation of particular imitations of those ideas. In a certain sense, Aristotle's method is both inductive and deductive, while Plato's is essentially deductive from a priori principles.

In the larger sense of the word, Aristotle makes philosophy coextensive with reasoning, which he also called 'science'. Note, however, that his use of the term science carries a different meaning than that which is covered by the scientific method. "All science (dianoia) is either practical, poetical or theoretical." By practical science he understands ethics and politics; by poetical, he means the study of poetry and the other fine arts; while by theoretical philosophy he means physics, mathematics, and metaphysics.

Aristotle's conception of logic was the dominant form of logic up until the advances in mathematical logic in the 19th century. Kant himself thought that Aristotle had done everything possible in terms of logic. The Organon is the name given by Aristotle's followers, the Peripatetics, for the standard collection of six of his works on logic. The system of logic described in two of these works, namely On Interpretation and the Prior Analytics, is often called Aristotelian logic.

Aristotle was the creator of syllogisms with modalities (modal logic). The word modal refers to the word 'modes', explaining the fact that modal logic deals with the modes of truth. Aristotle introduced the qualification of 'necessary' and 'possible' premises. He constructed a logic which helped in the evaluation of truth but which was difficult to interpret.

seen as essentially synonymous or perhaps more often logic is seen as the defining and pure form of reason.

However machines and animals can unconsciously perform logical operations, and many animals (including humans) can unconsciously, associate different perceptions as causes and effects and then make decisions or even plans. Therefore, to have any distinct meaning at all, 'reason' must be the type of thinking which links language, consciousness and logic, and at this time, only humans are known to combine these things.

However, note that reasoning is defined very differently depending on the context of the understanding of reason as a form of knowledge. The logical definition is the act of using reason to derive a conclusion from certain premises using a given methodology, and the two most commonly used explicit methods to reach a conclusion are deductive reasoning and inductive reasoning. However, within idealist philosophical contexts, reasoning is the mental process which informs our imagination, perceptions, thoughts, and feelings with whatever intelligibility these appear to contain; and thus links our experience with universal meaning. The specifics of the methods of reasoning are of interest to such disciplines as philosophy, logic, psychology, and artificial intelligence.

In deductive reasoning, given true premises, the conclusion must follow and it cannot be false. In this type of reasoning, the conclusion is inherent in the premises. Deductive reasoning therefore does not increase one's knowledge base and is said to be non–ampliative. Classic examples of deductive reasoning are found in such syllogisms as the following:

1. One must exist/live to perform the act of thinking.
2. I think.
3. Therefore, I am.

In inductive reasoning, on the other hand, when the premises are true, then the conclusion follows with some degree of *probability*.[5] This method of

[5] Recall that the word *probability* derives from the Latin 'probare' (to prove, or to test). Informally, probable is one of several words applied to uncertain events or knowledge, being closely related in meaning to likely, risky, hazardous, and doubtful. Chance, odds, and bet are other words expressing similar notions. Just as the theory of mechanics assigns precise definitions to such everyday terms as work and force, the theory of probability attempts to quantify the notion of probable.

The scientific study of probability is a modern development. Gambling shows that there has been an interest in quantifying the ideas of probability for millennia, but exact mathematical descriptions of use in those problems only arose much later. The doctrine of probabilities dates to the correspondence of Pierre de Fermat and Blaise Pascal (1654). Christiaan Huygens (1657) gave the earliest known scientific treatment of the subject. Jakob Bernoulli's 'Ars Conjectandi' (posthumous, 1713) and Abraham de Moivre's 'Doctrine of Chances' (1718) treated the subject as a branch of mathematics.

reasoning is ampliative, as it gives more information than what was contained in the premises themselves. A classical example comes from David Hume:[6]

1. The sun rose in the east every morning up until now.
2. Therefore the sun will also rise in the east tomorrow.

A third method of reasoning is called abductive reasoning, or inference to the best explanation. This method is more complex in its structure and can involve both inductive and deductive arguments. The main characteristic of abduction is that it is an attempt to favor one conclusion above others by either attempting to falsify alternative explanations, or showing the likelihood of the favored conclusion given a set of more or less disputable assumptions.

A fourth method of reasoning is analogy. Reasoning by analogy goes from a particular to another particular. The conclusion of an analogy is only plausible. Analogical reasoning is very frequent in common sense, science, philosophy and the humanities, but sometimes it is accepted only as an auxiliary method. A refined approach is *case–based reasoning*.

Pierre–Simon Laplace (1774) made the first attempt to deduce a rule for the combination of observations from the principles of the theory of probabilities. He represented the law of probability of errors by a curve $y = \varphi(x)$, x being any error and y its probability, and laid down three properties of this curve: (i) it is symmetric as to the $y-$axis; (ii) the $x-$axis is an asymptote, the probability of the error being 0; (iii) the area enclosed is 1, it being certain that an error exists. He deduced a formula for the *mean* of three observations. He also gave (1781) a formula for the law of facility of error (a term due to Lagrange, 1774), but one which led to unmanageable equations. Daniel Bernoulli (1778) introduced the principle of the maximum product of the probabilities of a system of concurrent errors.

The *method of least squares* is due to Adrien–Marie Legendre (1805), who introduced it in his 'Nouvelles méthodes pour la détermination des orbites des comètes' (New Methods for Determining the Orbits of Comets). In ignorance of Legendre's contribution, an Irish–American writer, Robert Adrain, editor of 'The Analyst' (1808), first deduced the law of facility of error,

$$\phi(x) = ce^{-h^2x^2}$$

where c and h are constants depending on precision of observation. He gave two proofs, the second being essentially the same as John Herschel's (1850). Carl Friedrich Gauss gave the first proof which seems to have been known in Europe (the third after Adrain's) in 1809. Further proofs were given by Laplace (1810, 1812), Gauss (1823), James Ivory (1825, 1826), Hagen (1837), Friedrich Bessel (1838), W. F. Donkin (1844, 1856), and Morgan Crofton (1870).

[6] David Hume (April 26, 1711 – August 25, 1776)[1] was a Scottish philosopher, economist, and historian, as well as an important figure of Western philosophy and of the Scottish Enlightenment.

Plan

Recall that a *plan* represents a proposed or intended method of getting from one set of circumstances to another. They are often used to move from the present situation, towards the achievement of one or more objectives or goals.

Informal or ad–hoc plans are created by individual humans in all of their pursuits. Structured and formal plans, used by multiple people, are more likely to occur in projects, diplomacy, careers, economic development, military campaigns, combat, or in the conduct of other business.

It is common for less formal plans to be created as abstract ideas, and remain in that form as they are maintained and put to use. More formal plans as used for business and military purposes, while initially created with and as an abstract thought, are likely to be written down, drawn up or otherwise stored in a form that is accessible to multiple people across time and space. This allows more reliable collaboration in the execution of the plan.

The term planning implies the working out of sub–components in some degree of detail. Broader–brush enunciations of objectives may qualify as metaphorical road–maps.

Planning literally just means the creation of a plan; it can be as simple as making a list. It has acquired a technical meaning, however, to cover the area of government legislation and regulations related to the use of resources.

Planning can refer to the planned use of any and all resources, as for example, in the succession of Five–Year Plans through which the government of the Soviet Union sought to develop the country. However, the term is most frequently used in relation to planning for the use of land and related resources, for example in urban planning, transportation planning, and so forth.

Problem Solving

The *problem solving* forms part of thinking. Considered the most complex of all intellectual functions, problem solving has been defined as higher–order cognitive process that requires the modulation and control of more routine or fundamental skills. It occurs if an organism or an artificial intelligence system does not know how to proceed from a given state to a desired goal state. It is part of the larger problem process that includes problem finding and problem shaping.

The nature of human problem solving has been studied by psychologists over the past hundred years. There are several methods of studying problem solving, including: *introspection*,[7] *behaviorism*,[8] computer simulation and experimental methods.

[7] Introspection is contemplation on one's self, as opposed to extrospection, the observation of things external to one's self. Introspection may be used synonymously with self–reflection and used in a similar way. Cognitive psychology accepts the use of the scientific method, but rejects introspection as a valid method

Beginning with the early experimental work of the Gestaltists in Germany (e.g., [Dun35], and continuing through the 1960s and early 1970s, research on problem solving typically conducted relatively simple, laboratory tasks that appeared novel to participants (see, e.g. [May92]). Various reasons account for the choice of simple novel tasks: they had clearly defined optimal solutions, they were solvable within a relatively short time frame, researchers could trace participants' problem–solving steps, and so on. The researchers made the underlying assumption, of course, that simple tasks such as the Tower of Hanoi captured the main properties of 'real world' problems, and that the cognitive processes underlying participants' attempts to solve simple problems were representative of the processes engaged in when solving 'real world' problems. Thus researchers used simple problems for reasons of convenience, and thought generalizations to more complex problems would become possible. Perhaps the best–known and most impressive example of this line of research remains the work by Newell and Simon [NS72].

See more on problem solving below.

Learning

Recall that learning is the process of acquiring knowledge, skills, attitudes, or values, through study, experience, or teaching, that causes a change of behavior that is persistent, measurable, and specified or allows an individual to formulate a new mental construct or revise a prior mental construct (conceptual knowledge such as attitudes or values). It is a process that depends on

of investigation. It should be noted that Herbert Simon and Allen Newell identified the 'thinking–aloud' protocol, in which investigators view a subject engaged in introspection, and who speaks his thoughts aloud, thus allowing study of his introspection.

Introspection was once an acceptable means of gaining insight into psychological phenomena. Introspection was used by German physiologist Wilhelm Wundt in the experimental psychology laboratory he had founded in Leipzig in 1879. Wundt believed that by using introspection in his experiments he would gather information into how the subject's minds were working, thus he wanted to examine the mind into its basic elements. Wundt did not invent this way of looking into an individual's mind through their experiences; rather, it can be dated back to Socrates. Wundt's distinctive contribution was to take this method into the experimental arena and thus into the newly formed field of psychology.

[8] Behaviorism is an approach to psychology based on the proposition that behavior can be studied and explained scientifically without recourse to internal mental states. A similar approach to political science may be found in Behavioralism. The behaviorist school of thought ran concurrent with the psychoanalysis movement in psychology in the 20th century. Its main influences were Ivan Pavlov, who investigated classical conditioning, John B. Watson who rejected introspective methods and sought to restrict psychology to experimental methods, and B.F. Skinner who conducted research on operant conditioning.

experience and leads to long–term changes in behavior potential. Behavior potential describes the possible behavior of an individual (not actual behavior) in a given situation in order to achieve a goal. But potential is not enough; if individual learning is not periodically reinforced, it becomes shallower and shallower, and eventually will be lost in that individual.

Short term changes in behavior potential, such as fatigue, do not constitute learning. Some long–term changes in behavior potential result from aging and development, rather than learning.

Education is the conscious attempt to promote learning in others. The primary function of 'teaching' is to create a safe, viable, productive learning environment. Management of the total learning environment to promote, enhance and motivate learning is a *paradigm shift*[9] from a focus on teaching to a focus on learning.

[9] Recall that an *epistemological paradigm shift* was called a *scientific revolution* by epistemologist and historian of science Thomas Kuhn in his 1962 book 'The Structure of Scientific Revolutions', to describe a change in basic assumptions within the ruling theory of science. It has since become widely applied to many other realms of human experience as well.

A scientific revolution occurs, according to Kuhn, when scientists encounter anomalies which cannot be explained by the universally accepted paradigm within which scientific progress has thereto been made. The paradigm, in Kuhn's view, is not simply the current theory, but the entire worldview in which it exists, and all of the implications which come with it. There are anomalies for all paradigms, Kuhn maintained, that are brushed away as acceptable levels of error, or simply ignored and not dealt with (a principal argument Kuhn uses to reject Karl Popper's model of falsifiability as the key force involved in scientific change). Rather, according to Kuhn, anomalies have various levels of significance to the practitioners of science at the time. To put it in the context of early 20th century physics, some scientists found the problems with calculating Mercury's perihelion more troubling than the Michelson–Morley experiment results, and some the other way around. Kuhn's model of scientific change differs here, and in many places, from that of the logical positivists in that it puts an enhanced emphasis on the individual humans involved as scientists, rather than abstracting science into a purely logical or philosophical venture. When enough significant anomalies have accrued against a current paradigm, the scientific discipline is thrown into a state of crisis, according to Kuhn. During this crisis, new ideas, perhaps ones previously discarded, are tried. Eventually a new paradigm is formed, which gains its own new followers, and an intellectual 'battle' takes place between the followers of the new paradigm and the hold–outs of the old paradigm. Again, for early 20th century physics, the transition between the Maxwellian electromagnetic worldview and the Einsteinian Relativistic worldview was not instantaneous nor calm, and instead involved a protracted set of 'attacks', both with empirical data as well as rhetorical or philosophical arguments, by both sides, with the Einsteinian theory winning out in the long–run. Again, the weighing of evidence and importance of new data was fit through the human sieve: some scientists found the simplicity

The stronger the stimulation for the brain, the deeper the impression that is left in the neuronal network. Therefore a repeated, very intensive experience perceived through all of the senses (audition, sight, smell) of an individual will remain longer and prevail over other experiences. The complex interactions of neurons that have formed a network in the brain determine the direction of flow of the micro–voltage electricity that flows through the brain when a person thinks. The characteristics of the neuronal network shaped by previous impressions is what we call the person's 'character'.

The most basic learning process is *imitation*, one's personal repetition of an observed process, such as a smile. Thus an imitation will take one's time (attention to the details), space (a location for learning), skills (or practice), and other resources (for example, a protected area). Through copying, most infants learn how to hunt (i.e., direct one's attention), feed and perform most basic tasks necessary for survival.

The so–called *Bloom's Taxonomy*[10] divides the learning process into a six–level hierarchy, where knowledge is the lowest order of cognition and evaluation the highest [Blo80]:

---

of Einstein's equations to be most compelling, while some found them more complicated than the notion of Maxwell's aether which they banished. Some found Eddington's photographs of light bending around the sun to be compelling, some questioned their accuracy and meaning. Sometimes the convincing force is just time itself and the human toll it takes, Kuhn pointed out, using a quote from Max Planck: "A new scientific truth does not triumph by convincing its opponents and making them see the light, but rather because its opponents eventually die, and a new generation grows up that is familiar with it." After a given discipline has changed from one paradigm to another, this is called, in Kuhn's terminology, a scientific revolution or a paradigm shift. It is often this final conclusion, the result of the long process, that is meant when the term paradigm shift is used colloquially: simply the (often radical) change of worldview, without reference to the specificities of Kuhn's historical argument.

[10]Benjamin Bloom (21 February 1913 – September 13, 1999) was an American educational psychologist who made significant contributions to the classification of educational objectives and the theory of mastery learning.
Bloom's classification of educational objectives, known as Bloom's Taxonomy, incorporates cognitive, psychomotor, and affective domains of knowledge. While working at the University of Chicago in the 1950s and '60s, he wrote two important books, Stability and Change in Human Characteristics and Taxonomy of Educational Objectives (1956). Bloom's taxonomy provides structure in which to categorize test questions. This taxonomy helps teachers pose questions in such a way to determine the level of understanding that a student possesses. For example, based upon the type of question asked, a teacher can determine that a student is competent in content knowledge, comprehension, application, analysis, synthesis and/or evaluation. This taxonomy is organized in a hierarchal way to organize information from basic factual recall to higher order thinking. This data table below is from the article written by W. Huitt titled, "Bloom *et al.*'s Taxonomy of

1. Knowledge is the memory of previously–learnt materials such as facts, terms, basic concepts and answers.
2. Comprehension is the understanding of facts and ideas by organization, comparison, translation, interpretation, and description.
3. Application is the use of new knowledge to solve problems.
4. Analysis is the examination and division of information into parts by identifying motives or causes. A person can analyze by making inferences and finding evidence to support generalizations.
5. Synthesis is the compilation of information in a new way by combining elements into patterns or proposing alternative solutions.
6. Evaluation is the presentation and defense of opinions by making judgments about information, validity of ideas or quality of work based on the following set of criteria:

- *Attention* – the cognitive process of selectively concentrating on one thing while ignoring other things. Examples include listening carefully to what someone is saying while ignoring other conversations in the room (e.g. the cocktail party problem, Cherry, 1953). Attention can also be split, as when a person drives a car and talks on a cell phone at the same time. Sometimes our attention shifts to matters unrelated to the external environment, this is referred to as mind-wandering or 'spontaneous thought'. Attention is one of the most intensely studied topics within psychology and cognitive neuroscience. Of the many cognitive processes associated with the human mind (decision–making, memory, emotion, etc), attention is considered the most concrete because it is tied so closely to perception. As such, it is a gateway to the rest of cognition. The most famous definition of attention was provided by one of the first major psychologists, William James[11] in

the Cognitive Domain". The table below describes the levels of Bloom's Taxonomy, beginning with the lowest level of basic factual recall. Each level in the table is defined, gives descriptive verbs that would foster each level of learning, and describes sample behaviors of that level. Bloom's taxonomy helps teachers better prepare questions that would foster basic knowledge recall all the way to questioning styles that foster synthesis and evaluation. By structuring the questioning format, teachers will be able to better understand what a child's weaknesses and strengths are and determine ways to help students think at a higher–level.

[11] William James (January 11, 1842 — August 26, 1910) was a pioneering American psychologist and philosopher. He wrote influential books on the young science of psychology, educational psychology, psychology of religious experience and mysticism, and the philosophy of pragmatism. He gained widespread recognition with his monumental Principles of 'Psychology' (1890), fourteen hundred pages in two volumes which took ten years to complete. 'Psychology: The Briefer Course', was an 1892 abridgement designed as a less rigorous introduction to the field. These works criticized both the English associationist school and the Hegelianism of

his 1890 book 'Principles of Psychology': "Everyone knows what attention is. It is the taking possession by the mind in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought ... It implies withdrawal from some things in order to deal effectively with others." Most experiments show that one neural correlate of attention is enhanced firing. Say a neuron has a certain response to a stimulus when the animal is not attending to that stimulus. When the animal attends to the stimulus, even if the physical characteristic of the stimulus remains the same the neurons response is enhanced. A strict criterion, in this paradigm of testing attention, is that the physical stimulus available to the subject must be the same, and only the mental state is allowed to change. In this manner, any differences in neuronal firing may be attributed to a mental state (attention) rather than differences in the stimulus itself.

- *Habituation* – an example of non–associative learning in which there is a progressive diminution of behavioral response probability with repetition of a stimulus. It is another form of integration. An animal first responds to a sensory stimulus, but if it is neither rewarding nor harmful the animal learns to suppress its response through repeated encounters. One example of this can be seen in small song birds – if a stuffed owl (or similar predator) is introduced into the cage, the birds react to it as though it were a real predator, but soon realise that it is not and so become habituated to it. If another stuffed owl is introduced (or the same one removed and re–introduced), the birds react to it as though it were a predator, showing that it is only a very specific stimulus that is being ignored (namely, one particular unmoving owl in one place). This learned suppression of response is habituation. Habituation is stimulus specific. It does not cause a general decline in responsiveness. It functions like an average weighted history wavelet interference filter reducing the responsiveness of the organism to a particular stimulus. Frequently one can see opponent processes after the

---

his day as competing dogmatisms of little explanatory value, and sought to re–conceive of the human mind as inherently purposive and selective.

James defined *true beliefs* as those that prove useful to the believer. Truth, he said, is that which works in the way of belief. "True ideas lead us into useful verbal and conceptual quarters as well as directly up to useful sensible termini. They lead to consistency, stability and flowing human intercourse" but "all true processes must lead to the face of directly verifying sensible experiences somewhere," he wrote.

Pragmatism as a view of the meaning of truth is considered obsolete by many in contemporary philosophy, because the predominant trend of thinking in the years since James' death in 1910 has been toward non–epistemic definitions of truth, i.e., definitions that don't make truth dependent upon the warrant of a belief. A contemporary philosopher or logician will often be found explaining that the statement 'the book is on the table' is true iff the book is on the table.

stimulus is removed. Habituation is connected to associational reciprocal inhibition phenomenon, opponent process, motion after effect, color constancy, size constancy, and negative image after effect. Habituation is frequently used in testing psychological phenomena. Both infants and adults look less and less as a result of consistent exposure to a particular stimulus. The amount of time spent looking to a presented alternate stimulus (after habituation to the initial stimulus) is indicative of the strength of the remembered percept of the previous stimulus. It is also used to discover the resolution of perceptual systems, for example, by habituating a subject to one stimulus, and then observing responses to similar ones, one can detect the smallest degree of difference that is detectable by the subject.

Closely related to habituation is *neural adaptation* or *sensory adaptation* – a change over time in the responsiveness of the sensory system to a constant stimulus. It is usually experienced as a change in the stimulus. For example, if one rests one's hand on a table, one immediately feels the table's surface on one's skin. Within a few seconds, however, one ceases to feel the table's surface. The sensory neurons stimulated by the table's surface respond immediately, but then respond less and less until they may not respond at all; this is neural adaptation. More generally, neural adaptation refers to a temporary change of the neural response to a stimulus as the result of preceding stimulation. It is usually distinguished from memory, which is thought to involve a more permanent change in neural responsiveness. Some people use adaptation as an umbrella term that encompasses the neural correlates of priming and habituation. In most cases, adaptation results in a response decrease, but response facilitation does also occur. Some adaptation may result from simple fatigue, but some may result from an active re–calibration of the responses of neurons to ensure optimal sensitivity. Adaptation is considered to be the cause of perceptual phenomena like afterimages and the motion aftereffect. In the absence of fixational eye movements, visual perception may fade out or disappear due to neural adaptation.

- *Sensitization* – an example of non–associative learning in which the progressive amplification of a response follows repeated administrations of a stimulus [BHB95]. For example, electrical or chemical stimulation of the rat hippocampus causes strengthening of synaptic signals, a process known as long–term potentiation (LTP). LTP is thought to underlie memory and learning in the human brain. A different type of sensitization is that of kindling, where repeated stimulation of hippocampal or amygdaloid neurons eventually leads to seizures. Thus, kindling has been suggested as a model for temporal lobe epilepsy. A third type is central sensitization, where nociceptive neurons in the dorsal horns of the spinal cord become sensitized

by peripheral tissue damage or inflammation. These various types indicate that sensitization may underlie both pathological and adaptive functions in the organism, but whether they also share the same physiological and molecular properties is not yet established.

- *Classical Pavlovian conditioning* – a type of associative learning. Ivan Pavlov described the learning of conditioned behavior as being formed by pairing two stimuli to condition an animal into giving a certain response. The simplest form of classical conditioning is reminiscent of what Aristotle would have called the law of contiguity, which states that: 'When two things commonly occur together, the appearance of one will bring the other to mind.' Classical conditioning focuses on reflexive behavior or involuntary behavior. Any reflex can be conditioned to respond to a formerly neutral stimulus. The typical paradigm for classical conditioning involves repeatedly pairing a neutral stimulus with an unconditioned stimulus. An unconditioned reflex is formed by an unconditioned stimulus, a stimulus that elicits a response–known as an unconditioned response–that is automatic and requires no learning and are usually apparent in all species. The relationship between the unconditioned stimulus and unconditioned response is known as the unconditioned reflex. The conditioned stimulus, is an initially neutral stimulus that elicits a response–known as a conditioned response–that is acquired through learning and can vary greatly amongst individuals. Conditioned stimuli are associated psychologically with conditions such as anticipation, satisfaction (both immediate and prolonged), and fear. The relationship between the conditioned stimulus and conditioned response is known as the conditioned (or conditional) reflex. In classical conditioning, when the unconditioned stimulus is repeatedly or strongly paired with a neutral stimulus the neutral stimulus becomes a conditioned stimulus and elicits a conditioned response.
- *Operant conditioning* – the use of consequences to modify the occurrence and form of behavior. Operant conditioning is distinguished from Pavlovian conditioning in that operant conditioning deals with the modification of voluntary behavior through the use of consequences, while Pavlovian conditioning deals with the conditioning of involuntary reflexive behavior so that it occurs under new antecedent conditions. Unlike reflexes, which are biologically fixed in form, the form of an operant response is modifiable by its consequences. Operant conditioning, sometimes called instrumental conditioning or instrumental learning, was first extensively studied by Edward Thorndike,[12] who observed the

[12] Edward Lee Thorndike (August 31, 1874 - August 9, 1949) was an American psychologist who spent nearly his entire career at Teachers College, Columbia University. His work on animal behavior and the learning process led to the theory of connectionism.

Among Thorndike's most famous contributions were his research on how cats learned to escape from puzzle boxes, and his related formulation of the law of effect. The law of effect states that responses which are closely followed by satisfy-

behavior of cats trying to escape from home-made puzzle boxes. When first constrained in the boxes, the cats took a long time to escape. With experience, ineffective responses occurred less frequently and successful responses occurred more frequently, enabling the cats to escape in less time over successive trials. In his Law of Effect, Thorndike theorized that successful responses, those producing satisfying consequences, were 'stamped in' by the experience and thus occurred more frequently. Unsuccessful responses, those producing annoying consequences, were stamped out and subsequently occurred less frequently. In short, some consequences strengthened behavior and some consequences weakened behavior. Burrhus Skinner[13] built upon Thorndike's ideas to construct a more detailed

---

ing consequences are associated with the situation, and are more likely to reoccur when the situation is subsequently encountered. Conversely, if the responses are followed by aversive consequences, associations to the situation become weaker. The puzzle box experiments were motivated in part by Thorndike's dislike for statements that animals made use of extraordinary faculties such as insight in their problem solving: "In the first place, most of the books do not give us a psychology, but rather a eulogy of animals. They have all been about animal intelligence, never about animal stupidity." (Animal Intelligence, 1911).

Thorndike meant to distinguish clearly whether or not cats escaping from puzzle boxes were using insight. Thorndike's instruments in answering this question were 'learning curves' revealed by plotting the time it took for an animal to escape the box each time it was in the box. He reasoned that if the animals were showing 'insight,' then their time to escape would suddenly drop to a negligible period, which would also be shown in the learning curve as an abrupt drop; while animals using a more ordinary method of trial and error would show gradual curves. His finding was that cats consistently showed gradual learning.

Thorndike interpreted the findings in terms of associations. He asserted that the connection between the box and the motions the cat used to escape was 'strengthened' by each escape. A similar, though radically reworked idea was taken up by B.F. Skinner in his formulation of Operant Conditioning, and the associative analysis went on to figure largely in behavioral work through mid-century, now evident in some modern work in behavior as well as modern *connectionism*.

[13] Burrhus Frederic Skinner (March 20, 1904 – August 18, 1990) was an American psychologist and author. He conducted pioneering work on experimental psychology and advocated behaviorism, which seeks to understand behavior as a function of environmental histories of experiencing consequences. He also wrote a number of controversial works in which he proposed the widespread use of psychological behavior modification techniques, primarily operant conditioning, in order to improve society and increase human happiness; and as a form of social engineering.

Skinner was born in rural Susquehanna, Pennsylvania. He attended Hamilton College in New York with the intention of becoming a writer and received a B.A. in English literature in 1926. After graduation, he spent a year in Greenwich Village attempting to become a writer of fiction, but he soon became disillusioned with his literary skills and concluded that he had little world experience, and no

theory of operant conditioning based on: (a) *reinforcement* (a consequence that causes a behavior to occur with greater frequency), (b) *punishment* (a consequence that causes a behavior to occur with less frequency), and (c) *extinction* (the lack of any consequence following a response). There are four contexts of operant conditioning:
(i) *Positive reinforcement* occurs when a behavior (response) is followed by a favorable stimulus (commonly seen as pleasant) that increases the frequency of that behavior. In the Skinner box experiment, a stimulus such as food or sugar solution can be delivered when the rat engages in a target behavior, such as pressing a lever.
(ii) *Negative reinforcement* occurs when a behavior (response) is followed by the removal of an aversive stimulus (commonly seen as unpleasant) thereby increasing that behavior's frequency. In the Skinner box experiment, negative reinforcement can be a loud noise continuously sounding inside the rat's cage until it engages in the target behavior, such as pressing a lever, upon which the loud noise is removed.
(iii) *Positive punishment* (also called 'Punishment by contingent stimulation') occurs when a behavior (response) is followed by an aversive stimulus, such as introducing a shock or loud noise, resulting in a decrease in that behavior.
(iv) *Negative punishment* (also called 'Punishment by contingent withdrawal') occurs when a behavior (response) is followed by the removal of a favorable stimulus, such as taking away a child's toy following an undesired behavior, resulting in a decrease in that behavior.

- *Observational (or social) learning* – learning that occurs as a function of observing, retaining and replicating behavior observed in others. It is most associated with the work of psychologist Albert Bandura,[14] who implemented some of the seminal studies in the area and initiated social learning theory. Although observational learning can take place at any stage in life, it is thought to be particularly important during childhood, particularly as authority becomes important. Because of this, social learning theory has influenced debates on the effect of television violence and parental role models. Bandura's Bobo doll experiment is widely cited in

---

strong personal perspective from which to write. During this time, which Skinner later called 'the dark year,' he chanced upon a copy of Bertrand Russell's book 'An Outline of Philosophy', in which Russell discusses the behaviorist philosophy of psychologist John B. Watson. At the time, Skinner had begun to take more interest in the actions and behaviors of those around him, and some of his short stories had taken a 'psychological' slant. He decided to abandon literature and seek admission as a graduate student in psychology at Harvard University (which at the time was not regarded as a leading institution in that field).

[14] Albert Bandura (born December 4, 1925 in Mundare, Alberta) is a Canadian psychologist most famous for his work on social learning theory (or Social Cognitivism) and self efficacy. He is particularly noted for the Bobo doll experiment.

psychology as a demonstration of observational learning and demonstrated that children are more likely to engage in violent play with a life size rebounding doll after watching an adult do the same. Observational learning allows for learning without any change in behavior and has therefore been used as an argument against strict behaviorism which argued that behavior change must occur for new behaviors to be acquired. Bandura called the process of social learning modelling and gave four conditions required for a person to successfully model the behavior of someone else: (i) attention to the model (a person must first pay attention to a person engaging in a certain behavior – the model); (ii) retention of details (once attending to the observed behavior, the observer must be able to effectively remember what the model has done); (iii) motor reproduction (the observer must be able to replicate the behavior being observed; e.g., juggling cannot be effectively learned by observing a model juggler if the observer does not already have the ability to perform the component actions, i.e., throwing and catching a ball); (iv) motivation and opportunity (the observer must be motivated to carry out the action they have observed and remembered, and must have the opportunity to do so; e.g., a suitably skilled person must want to replicate the behavior of a model juggler, and needs to have an appropriate number of items to juggle to hand). Social learning may affect behavior in the following ways: (i) teaches new behaviors; (ii) increases or decreases the frequency with which previously learned behaviors are carried out; (iii) can encourage previously forbidden behaviors; (iv) can increase or decrease similar behaviors (e.g., observing a model excelling in piano playing may encourage an observer to excel in playing the saxophone).

- *Communication* – the process of symbolic activity, sometimes via a language. Specialized fields focus on various aspects of communication, and include: (i) *mass communication* (academic study of various means by which individuals and entities relay information to large segments of the population all at once through mass media); (ii) *communication studies* (academic discipline that studies communication; subdisciplines include argumentation, speech communication, rhetoric, communication theory, performance studies, group communication, information theory, intercultural communication, interpersonal communication, intrapersonal communication, marketing, organizational communication, persuasion, propaganda, public affairs, public relations and telecommunication); (iii) *organizational communication* (the study of how people communicate within an organizational context, or the influence of, or interaction with organizational structures in communicating/organizing), (iv) *conversation analysis* (commonly abbreviated as CA, is the study of talk in interaction; CA generally attempts to describe the orderliness, structure and sequential patterns of interaction, whether this is institutional, in the school, doctor's

surgery, courts or elsewhere, or casual conversation); (v) *linguistics* (scientific study of human language and speech; usually is conducted along two major axes: theoretical vs. applied, and autonomous vs. contextual); (vi) *cognitive linguistics* (commonly abbreviated as CL, refers to the school of linguistics that views the important essence of language as innately based in evolutionary–developed and speciated faculties, and seeks explanations that advance or fit well into the current understandings of the human mind); (vii) *sociolinguistics* (the study of the effect of any and all aspects of society, including cultural norms, expectations, and context, on the way language is used); (viii) *pragmatics* (concerned with bridging the explanatory gap between sentence meaning and speaker's meaning – how context influences the interpretation is crucial); (ix) *semiotics* (the study of signs, both individually and grouped in sign systems; it includes the study of how meaning is made and understood); and (x) *discourse analysis* (a general term for a number of approaches to analyzing written, spoken or signed language use; includes: discourse grammar, rhetoric and stylistics). Communication as a named and unified discipline has a history of contestation that goes back to the Socratic dialogues, in many ways making it the first and most contestatory of all early sciences and philosophies. Seeking to define 'communication' as a static word or unified discipline may not be as important as understanding communication as a family of resemblances with a plurality of definitions as Ludwig Wittgenstein[15] had put forth. Some definitions are broad, recognizing that animals can communicate, and some are more narrow, only including human beings within the parameters of human symbolic interaction. Nonetheless, communication is usually described along three major dimensions: content, form, and destination. In the advent of 'noise' (internal psychological noise and/or physical realities) these three components of communication often become skewed and inaccurate. (between parties, communication content include acts that declare knowledge and experiences, give advice and commands, and ask questions. These acts may take many forms, including gestures (nonverbal communication, sign language and body language), writing, or verbal speaking. The form depends on the symbol systems used. Together, communication content and form make messages that are sent towards a destination. The target can be oneself, another person (in interpersonal communication), or another entity (such as a corporation or group). There

[15] Ludwig Josef Johann Wittgenstein (April 26, 1889 – April 29, 1951) was an Austrian philosopher who contributed several ground-breaking works to contemporary philosophy, primarily on the foundations of logic, the philosophy of mathematics, the philosophy of language, and the philosophy of mind. He is widely regarded as one of the most influential philosophers of the 20th century.

are many theories of communication, and a commonly held assumption is that communication must be directed towards another person or entity. This essentially ignores intrapersonal communication (note intra–, not inter–) via diaries or self–talk. Interpersonal conversation can occur in dyads and groups of various sizes, and the size of the group impacts the nature of the talk. Small–group communication takes place in settings of between three and 12 individuals, and differs from large group interaction in companies or communities. This form of communication formed by a dyad and larger is sometimes referred to as the psychological model of communication where in a message is sent by a sender through channel to a receiver. At the largest level, mass communication describes messages sent to huge numbers of individuals through mass media, although there is debate if this is an interpersonal conversation.

Language

Recall that a language is a system of signals, such as voice sounds, gestures or written symbols that encode or decode information.

Human spoken and written languages can be described as a system of symbols (sometimes known as lexemes) and the grammars (rules) by which the symbols are manipulated. The word 'language' is also used to refer to common properties of languages.

Language learning is normal in human childhood. Most human languages use patterns of sound or gesture for symbols which enable communication with others around them. There are thousands of human languages, and these seem to share certain properties, even though many shared properties have exceptions.

Languages are not just sets of symbols. They also often conform to a rough grammar, or system of rules, used to manipulate the symbols. While a set of symbols may be used for expression or communication, it is primitive and relatively unexpressive, because there are no clear or regular relationships between the symbols.

Human languages are usually referred to as natural languages, and the science of studying them is *linguistics*, with Ferdinand de Saussure[16] and Noam Chomsky[17] as the most influential figures.

[16] Ferdinand de Saussure (November 26, 1857 – February 22, 1913) was a Geneva–born Swiss linguist whose ideas laid the foundation for many of the significant developments in linguistics in the 20th century. He is widely considered the 'father' of 20th–century linguistics.

Saussure's most influential work, 'Course in General Linguistics', was published posthumously in 1916 by former students Charles Bally and Albert Sechehaye on the basis of notes taken from Saussure's lectures at the University of Geneva. The Course became one of the seminal linguistics works of the 20th century, not primarily for the content (many of the ideas had been anticipated in the works

Humans and computer programs have also constructed other languages, including constructed languages such as Esperanto, Ido, Interlingua, Klingon, programming languages, and various mathematical formalisms. These

---

of other 19th century linguists), but rather for the innovative approach that Saussure applied in discussing linguistic phenomena. Its central notion is that language may be analyzed as a formal system of differential elements, apart from the messy dialectics of realtime production and comprehension.

Saussure's famous quotes are:

"A sign is the basic unit of language (a given language at a given time). Every language is a complete system of signs. Parole (the speech of an individual) is an external manifestation of language."

"A linguistic system is a series of differences of sound combined with a series of differences of ideas."

[17] Noam Avram Chomsky (born December 7, 1928) is the Institute Professor Emeritus of linguistics at the MIT. Chomsky is credited with the creation of the theory of generative grammar, considered to be one of the most significant contributions to the field of theoretical linguistics made in the 20th century. He also helped spark the cognitive revolution in psychology through his review of B.F. Skinner's 'Verbal Behavior', in which he challenged the behaviorist approach to the study of mind and language dominant in the 1950s. His naturalistic approach to the study of language has also affected the philosophy of language and mind. He is also credited with the establishment of the so–called *Chomsky hierarchy*, a classification of formal languages in terms of their generative power.

'Syntactic Structures' was a distillation of Chomsky's book 'Logical Structure of Linguistic Theory' (1955) in which he introduces transformational grammars. The theory takes utterances (sequences of words) to have a syntax which can be (largely) characterised by a formal grammar; in particular, a *context–free grammar* extended with transformational rules. Children are hypothesised to have an innate knowledge of the basic grammatical structure common to all human languages (i.e. they assume that any language which they encounter is of a certain restricted kind). This innate knowledge is often referred to as universal grammar. It is argued that modelling knowledge of language using a formal grammar accounts for the 'productivity' of language: with a limited set of grammar rules and a finite set of terms, humans are able to produce an infinite number of sentences, including sentences no one has previously said.

Chomsky's ideas have had a strong influence on researchers investigating the acquisition of language in children, though some researchers who work in this area today do not support Chomsky's theories, often advocating emergentist or connectionist theories reducing language to an instance of general processing mechanisms in the brain.

Chomsky's work in linguistics has had major implications for modern psychology. For Chomsky linguistics is a branch of cognitive psychology; genuine insights in linguistics imply concomitant understandings of aspects of mental processing and human nature. His theory of a universal grammar was seen by many as a direct challenge to the established behaviorist theories of the time and had major consequences for understanding how language is learned by children and what,

languages are not necessarily restricted to the properties shared by human languages.

Some of the areas of the human brain involved in language processing are: Broca's area, Wernicke's area, Supramarginal gyrus, Angular gyrus, Primary Auditory Cortex.

Mathematics and computer science use artificial entities called *formal languages* (including programming languages and markup languages, but also some that are far more theoretical in nature). These often take the form of character strings, produced by some combination of formal grammar and semantics of arbitrary complexity.

The classification of natural languages can be performed on the basis of different underlying principles (different closeness notions, respecting different properties and relations between languages); important directions of present classifications are:

1. Paying attention to the historical evolution of languages results in a genetic classification of languages—which is based on genetic relatedness of languages;
2. Paying attention to the internal structure of languages (grammar) results in a typological classification of languages—which is based on similarity of one or more components of the language's grammar across languages; and
3. Respecting geographical closeness and contacts between language-speaking communities results in areal groupings of languages.
4. The different classifications do not match each other and are not expected to, but the correlation between them is an important point for many linguistic research works. (Note that there is a parallel to the classification of species in biological phylogenetics here: consider monophyletic vs. polyphyletic groups of species.)

The task of genetic classification belongs to the field of historical–comparative linguistics, of typological—to linguistic typology. The world's languages have been grouped into families of languages that are believed to have common ancestors. Some of the major families are the Indo–European languages, the Afro–Asiatic languages, the Austronesian languages, and the Sino–Tibetan languages. The shared features of languages from one family can be due to shared ancestry.

An example of a typological classification is the classification of languages on the basis of the basic order of the verb, the subject and the object in a sentence into several types: SVO, SOV, VSO, and so on, languages. (English, for instance, belongs to the SVO language type.)

---

exactly, is the ability to use language. Many of the more basic principles of this theory (though not necessarily the stronger claims made by the principles and parameters approach described above) are now generally accepted in some circles.

The shared features of languages of one type (= from one typological class) may have arisen completely independently. (Compare with analogy in biology.) Their cooccurence might be due to the universal laws governing the structure of natural languages—language universals.

The following language groupings can serve as some linguistically significant examples of areal linguistic units, or sprachbunds: Balkan linguistic union, or the bigger group of European languages; Caucasian languages. Although the members of each group are not closely genetically related, there is a reason for them to share similar features, namely: their speakers have been in contact for a long time within a common community and the languages converged in the course of the history. These are called 'areal features'.

Mathematics and computer science use artificial entities called formal languages (including programming languages and markup languages, but also some that are far more theoretical in nature). These often take the form of character strings, produced by some combination of formal grammar and semantics of arbitrary complexity.

Abstraction

Recall that *abstraction* is the process of reducing the information content of a concept, typically in order to retain only information which is relevant for a particular purpose. For example, abstracting a leather soccer ball to a ball retains only the information on general ball attributes and behavior. Similarly, abstracting an emotional state to happiness reduces the amount of information conveyed about the emotional state.

Abstraction typically results in complexity reduction leading to a simpler conceptualization of a domain in order to facilitate processing or understanding of many specific scenarios in a generic way.

In philosophical terminology, abstraction is the thought process wherein ideas are distanced from objects.

Abstraction uses a strategy of simplification, wherein formerly concrete details are left ambiguous, vague, or undefined; thus effective communication about things in the abstract requires an intuitive or common experience between the communicator and the communication recipient.

Abstractions sometimes have ambiguous referents; for example, 'happiness' (when used as an abstraction) can refer to as many things as there are people and events or states of being which make them happy. Likewise, 'architecture' refers not only to the design of safe, functional buildings, but also to elements of creation and innovation which aim at elegant solutions to construction problems, to the use of space, and at its best, to the attempt to evoke an emotional response in the builders, owners, viewers and users of the building.

Abstraction in philosophy is the process (or, to some, the alleged process) in concept–formation of recognizing some set of common features in individuals, and on that basis forming a concept of that feature. The notion of abstraction is important to understanding some philosophical controversies surrounding

empiricism and the problem of universals. It has also recently become popular in formal logic under predicate abstraction.

Some research into the human brain suggests that the left and right hemispheres differ in their handling of abstraction. One side handles collections of examples (e.g., examples of a tree) whereas the other handles the concept itself.

Abstraction in mathematics is the process of extracting the underlying essence of a mathematical concept, removing any dependence on real world objects with which it might originally have been connected, and generalizing it so that it has wider applications.

Many areas of mathematics began with the study of real world problems, before the underlying rules and concepts were identified and defined as abstract structures. For example, geometry has its origins in the calculation of distances and areas in the real world; statistics has its origins in the calculation of probabilities in gambling; and algebra started with methods of solving problems in arithmetic.

Abstraction is an ongoing process in mathematics and the historical development of many mathematical topics exhibits a progression from the concrete to the abstract. Take the historical development of geometry as an example; the first steps in the abstraction of geometry were made by the ancient Greeks, with Euclid being the first person (as far as we know) to document the axioms of plane geometry. In the 17th century Descartes introduced Cartesian coordinates which allowed the development of analytic geometry. Further steps in abstraction were taken by Lobachevsky, Bolyai and Gauss[18]

[18] *Gauss–Bolyai–Lobachevsky space* is a non–Euclidean space with a negative Gaussian curvature, that is, a *hyperbolic geometry*. The main topic of conversation involving Gauss–Bolyai–Lobachevsky space involves the impossible process (at least in Euclidean geometry) of squaring the circle. The space is named after Carl Gauss, János Bolyai, and Nikolai Lobachevsky.

Carl Friedrich Gauss (30 April 1777 – 23 February 1855) was a German mathematician and scientist of profound genius who contributed significantly to many fields, including number theory, analysis, differential geometry, geodesy, magnetism, astronomy and optics. Sometimes known as 'the prince of mathematicians' and 'greatest mathematician since antiquity', Gauss had a remarkable influence in many fields of mathematics and science and is ranked among one of history's most influential mathematicians. Gauss was a child prodigy, of whom there are many anecdotes pertaining to his astounding precocity while a mere toddler, and made his first ground–breaking mathematical discoveries while still a teenager. He completed Disquisitiones Arithmeticae, his magnum opus, at the age of twenty–one (1798), though it would not be published until 1801. This work was fundamental in consolidating number theory as a discipline and has shaped the field to the present day. One of his most important results is his 'Theorema Egregrium', establishing an important property of the notion of curvature as a foundation of differential geometry.

János Bolyai (December 15, 1802–January 27, 1860) was a Hungarian mathematician. Between 1820 and 1823 he prepared a treatise on a complete system

who generalized the concepts of geometry to develop non–Euclidean geometries. Later in the 19th century mathematicians generalized geometry even further, developing such areas as geometry in n dimensions, projective geometry, affine geometry, finite geometry and differential geometry. Finally Felix Klein's 'Erlangen program'[19] identified the underlying theme of all of these geometries, defining each of them as the study of properties invariant under a given group of symmetries. This level of abstraction revealed deep connections between geometry and abstract algebra.

The advantages of abstraction are:

(i) It reveals deep connections between different areas of mathematics;

(ii) Known results in one area can suggest conjectures in a related area; and

(iii) Techniques and methods from one area can be applied to prove results in a related area.

An abstract structure is a formal object that is defined by a set of laws, properties, and relationships in a way that is logically if not always historically independent of the structure of contingent experiences, for example, those involving physical objects. Abstract structures are studied not only in logic and mathematics but in the fields that apply them, as computer science, and in the studies that reflect on them, as philosophy and especially the philosophy of mathematics. Indeed, modern mathematics has been defined in a very general sense as the study of abstract structures by the *Bourbaki* group.[20]

---

of non–Euclidean geometry. Bolyai's work was published in 1832 as an appendix to a mathematics textbook by his father. Gauss, on reading the Appendix, wrote to a friend saying "I regard this young geometer Bolyai as a genius of the first order." In 1848 Bolyai discovered not only that Lobachevsky had published a similar piece of work in 1829, but also a generalisation of this theory.

Nikolai Ivanovich Lobachevsky (December 1, 1792–February 24, 1856 (N.S.) was a Russian mathematician. Lobachevsky's main achievement is the development (independently from János Bolyai) of non–Euclidean geometry. Before him, mathematicians were trying to deduce Euclid's fifth postulate from other axioms. Lobachevsky would instead develop a geometry in which the fifth postulate was not true.

[19] Felix Christian Klein (April 25, 1849, Düsseldorf, Germany – June 22, 1925, Göttingen) was a German mathematician, known for his work in group theory, function theory, non-Euclidean geometry, and on the connections between geometry and group theory. His 1872 Erlangen Program, classifying geometries by their underlying symmetry groups, was a hugely influential synthesis of much of the mathematics of the day.

[20] Nicolas Bourbaki is the collective allonym under which a group of (mainly French) 20th-century mathematicians wrote a series of books presenting an exposition of modern advanced mathematics, beginning in 1935. With the goal of founding all of mathematics on set theory, the group strove for utmost rigour and generality, creating some new terminology and concepts along the way.

While Nicolas Bourbaki is an invented personage, the Bourbaki group is officially known as the Association des collaborateurs de Nicolas Bourbaki

The main disadvantage of abstraction is that highly abstract concepts are more difficult to learn, and require a degree of mathematical maturity and experience before they can be assimilated.

In computer science, abstraction is a mechanism and practice to reduce and factor out details so that one can focus on a few concepts at a time.

The concept is by analogy with abstraction in mathematics. The mathematical technique of abstraction begins with mathematical definitions; this has the fortunate effect of finessing some of the vexing philosophical issues of abstraction. For example, in both computing and in mathematics, numbers are concepts in the programming languages, as founded in mathematics. Implementation details depend on the hardware and software, but this is not a restriction because the computing concept of number is still based on the mathematical concept.

Roughly speaking, abstraction can be either that of control or data. Control abstraction is the abstraction of actions while data abstraction is that of data structures. For example, control abstraction in structured programming is the use of subprograms and formatted control flows. Data abstraction is to allow for handling data bits in meaningful manners. For example, it is the basic motivation behind data–type. Object–oriented programming can be seen as an attempt to abstract both data and code.

Creativity

Now, recall that *creativity* is a mental process involving the generation of new ideas or concepts, or new associations between existing ideas or concepts. From a scientific point of view, the products of creative thought (sometimes referred to as divergent thought) are usually considered to have both originality and

---

('association of collaborators of Nicolas Bourbaki'), which has an office at the École Normale Supérieure in Paris.

The emphasis on rigour may be seen as a reaction to the work of Jules–Henri Poincaré, who stressed the importance of free–flowing mathematical intuition, at a cost in completeness (i.e., proof) in presentation. The impact of Bourbaki's work initially was great on many active research mathematicians world–wide.

Notations introduced by Bourbaki include: the symbol $\varnothing$ for the *empty set*, and the terms *injective*, *surjective*, and *bijective*.

Aiming at a completely self-contained treatment of most of modern mathematics based on set theory, the group produced the following volumes (with the original French titles in parentheses):

I Set theory (Théorie des ensembles);
II Algebra (Algèbre);
III General Topology (Topologie générale);
IV Functions of one real variable (Fonctions d'une variable réelle);
V Topological vector spaces (Espaces vectoriels topologiques);
VI Integration (Intégration);
VII Commutative algebra (Algèbre commutative); and
VIII Lie groups and algebras (Groupes et algèbres de Lie).

appropriateness. An alternative, more everyday conception of creativity is that it is simply the act of making something new. Although intuitively a simple phenomenon, it is in fact quite complex. It has been studied from the perspectives of behavioral psychology, social psychology, psychometrics, cognitive science, artificial intelligence, philosophy, history, economics, design research, business, and management, among others. The studies have covered everyday creativity, exceptional creativity and even artificial creativity. Unlike many phenomena in science, there is no single, authoritative perspective or definition of creativity. Unlike many phenomena in psychology, there is no standardized measurement technique.

Creativity has been attributed variously to divine intervention, cognitive processes, the social environment, personality traits, and chance ('accident', 'serendipity'). It has been associated with genius, mental illness and humor. Some say it is a trait we are born with; others say it can be taught with the application of simple techniques. Although popularly associated with art and literature, it is also an essential part of innovation and invention and is important in professions such as business, economics, architecture, industrial design, science and engineering.

Despite, or perhaps because of, the ambiguity and multi–dimensional nature of creativity, entire industries have been spawned from the pursuit of creative ideas and the development of creativity techniques. This mysterious phenomenon, though undeniably important and constantly visible, seems to lie tantalizingly beyond the grasp of scientific investigation.

More than 60 different definitions of creativity can be found in the psychological literature (see [Tay88]). The etymological root of the word in English and most other European languages comes from the Latin 'creatus', which literally means 'to have grown'. Perhaps the most widespread conception of creativity in the scholarly literature is that creativity is manifested in the production of a creative work (for example, a new work of art or a scientific hypothesis) that is both novel and useful. Colloquial definitions of creativity are typically descriptive of activity that results in producing or bringing about something partly or wholly new; in investing an existing object with new properties or characteristics; in imagining new possibilities that were not conceived of before; and in seeing or performing something in a manner different from what was thought possible or normal previously.

A useful distinction has been made by [Rho61], between the creative person, the creative product, the creative process, and the creative 'press' or environment. Each of these factors are usually present in creative activity. This has been elaborated by [Joh72], who suggested that creative activity may exhibit several dimensions including sensitivity to problems on the part of the creative agent, originality, ingenuity, unusualness, usefulness, and appropriateness in relation to the creative product, and intellectual leadership on the part of the *creative agent*.

Boden [Bod04] noted that it is important to distinguish between ideas which are psychologically creative (which are novel to the individual mind

which had the idea), and those which are historically creative (which are novel with respect to the whole of human history). Drawing on ideas from artificial intelligence, she defines psychologically creative ideas as those which cannot be produced by the same set of generative rules as other, familiar ideas.

Often implied in the notion of creativity is a concomitant presence of inspiration, cognitive leaps, or intuitive insight as a part of creative thought and action [Koe64]. Popular psychology sometimes associates creativity with right or forehead brain activity or even specifically with lateral thinking. Some students of creativity have emphasized an element of chance in the creative process. Linus Pauling,[21] asked at a public lecture how one creates scientific theories, replied that one must endeavor to come up with many ideas — then discard the useless ones.

The formal starting point of the scientific study of creativity is sometimes considered to be J. Joy Guilford's[22] address to the American Psychological Association in 1950, which helped to popularize the topic (see [SL99]). Since then, researchers from a variety of fields have studied the nature of creativity

[21] Linus Carl Pauling (February 28, 1901 – August 19, 1994) was an American quantum chemist and biochemist, widely regarded as the premier chemist of the twentieth century. Pauling was a pioneer in the application of quantum mechanics to chemistry (quantum mechanics can, in principle, describe all of chemistry and molecular biology), and in 1954 was awarded the Nobel Prize in chemistry for his work describing the nature of chemical bonds. He also made important contributions to crystal and protein structure determination, and was one of the founders of molecular biology. Pauling is noted as a versatile scholar for his expertise in inorganic chemistry, organic chemistry, metallurgy, immunology, anesthesiology, psychology, debate, radioactive decay, and the aftermath of nuclear weapons, in addition to quantum mechanics and molecular biology.

Pauling received the Nobel Peace Prize in 1962 for his campaign against aboveground nuclear testing, becoming the only person in history to individually receive two Nobel Prizes (Marie Curie won Nobel Prizes in physics and chemistry, but shared the former and won the latter individually; John Bardeen won two Nobel Prizes in the field of physics, but both were shared; Frederick Sanger won two Nobel Prizes in chemistry, but one was shared).

Later in life, he became an advocate for regular consumption of massive doses of vitamin C, which is still regarded as unorthodox by conventional medicine.

[22] Joy Paul Guilford (1897–1988) was a US psychologist, best remembered for his psychometric study of human intelligence.

He graduated from the University of Nebraska before studying under Edward Titchener at Cornell. He then held a number of posts at Nebraska and briefly at the University of Southern California before becoming Director of Psychological Research at Santa Ana Army Air Base in 1941. There he worked on the selection and ranking of air–crew trainees.

Developing the views of L. L. Thurstone, Guilford rejected Charles Spearman's view that intelligence could be characterized in a single numerical parameter and proposed that three dimensions were necessary for accurate description: (i) content, (ii) operations, and (iii) productions. He made the important distinction between convergent and divergent production.

from a scientific point of view. Others have taken a more pragmatic approach, teaching practical creativity techniques. Three of the best–known are Alex Osborn's[23] *brainstorming* techniques, Genrikh Altshuller's[24] 'Theory of Inventive Problem Solving' (TIPS), and Edward de Bono's[25] *lateral thinking* (1960s to present).

The *neurology of creativity* has been discussed by F. Balzac in [Bal06]. The study found that creative innovation requires *coactivation and communication between regions of the brain that ordinarily are not strongly connected*. Highly creative people who excel at creative innovation tend to differ from others in three ways: they have a high level of specialized knowledge, they are capable of divergent thinking mediated by the frontal lobe, and they are able to modulate neurotransmitters such as norepinephrine in their frontal lobe. Thus, the frontal lobe appears to be the part of the cortex that is most important for creativity. The study also explored the links between creativity and sleep, mood and addiction disorders, and depression.

J. Guilford's group developed the so–called 'Torrance Tests of Creative Thinking'. They involved simple tests of divergent thinking and other problem–solving skills, which were scored on [Gui67]:

1. Fluency: the total number of interpretable, meaningful, and relevant ideas generated in response to the stimulus;
2. Flexibility: the number of different categories of relevant responses;

[23] Alex Faickney Osborn (May 24, 1888 – May 4, 1966) was an advertising manager and the author of the creativity technique named *brainstorming*.

[24] Genrikh Saulovich Altshuller (October 15, 1926 - September 24, 1998), created the Theory of Inventive Problem Solving (TIPS). Working as a clerk in a patent office, Altshuller embarked on finding some generic rules that would explain creation of new, inventive, patentable ideas.

[25] Edward de Bono (born May 19, 1933) is a psychologist and physician. De Bono writes prolifically on subjects of lateral thinking, a concept he is believed to have pioneered and now holds training seminars in. Dr. de Bono is also a world-famous consultant who has worked with companies like Coca-cola and Ericsson. In 1979 he co–founded the School of Thinking with Dr Michael Hewitt–Gleeson.

De Bono has detailed a range of 'deliberate thinking methods' – applications emphasizing thinking as a deliberate act rather than a reactive one. His writing style is simple and clear, though often criticized for being dry and repetitive. Avoiding academic terminology, he has advanced applied psychology by making theories about creativity and perception into usable tools. A distinctive feature of De Bono's books is that he never acknowledges or credits the ideas of other authors or researchers in the field of creativity.

De Bono's work has become particularly popular in the sphere of business – perhaps because of the perceived need to restructure corporations, to allow more flexible working practices and to innovate in products and services. The methods have migrated into corporate training courses designed to help employees and executives 'think out of the box' / 'think outside the box'.

3. Originality: the statistical rarity of the responses among the test subjects; and
4. Elaboration: the amount of detail in the responses.

Personality

On the other hand, *personality* is a *collection of emotional, thought and behavioral patterns unique to a person* that is consistent over time. Personality psychology is a branch of psychology which studies personality and individual different processes – that which makes us into a person. One emphasis is on trying to create a coherent picture of a person and all his or her major psychological processes. Another emphasis views it as the study of individual differences. These two views work together in practice. Personality psychologists are interested in broad view of the individual. This often leads to an interest in the most salient individual differences among people.

The word *personality* originates from the Latin *persona*, which means 'mask'.[26] In the History of theater of the ancient Latin world, the mask was not used as a plot device to disguise the identity of a character, but rather was a convention employed to represent, or typify that character.

There are several theoretical perspectives on personality in psychology, which involve different ideas about the relationship between personality and other psychological constructs, as well as different theories about the way personality develops. Most theories can be grouped into one of the following classes.

Generally the opponents to personality theories claim that personality is 'plastic' in time, places, moods and situations. Changing personality may in fact resulting from diet (or lack of), medical effects, historical or subsequent events, or learning. Stage managers (of many types) are especially skilled in changing a person's resulting 'personality'. Most personality theories will not cover such flexible nor unusual people situations. Therefore, although personality theories do not define personality as 'plastic' over time like their opponents, they do imply a drastic change in personality is highly unusual.

According to the Diagnostic and Statistical Manual of Mental Disorders of the American Psychiatric Association, personality traits are 'prominent aspects of personality that are exhibited in a wide range of important social and personal contexts.' In other words, persons have certain characteristics which partly determine their behavior. According to the theory, a friendly

[26] A *persona*, in the word's everyday usage, is a social role, or a character played by an actor. The word derives from the Latin for 'mask' or 'character', derived from the Etruscan word 'phersu', with the same meaning.

For instance, in Dostoevsky's novel, Notes from Underground (generally considered to be the first existentialist novel), the narrator ought not to be conflated with Dostoevsky himself, despite the fact that Dostoevsky and his narrator may or may not have shared much in common. In this sense, the persona is basically a mouthpiece for a particular world–view.

person is likely to act friendly in any situation because of the traits in his personality. One criticism of trait models of personality as a whole is that they lead professionals in clinical psychology and lay–people alike to accept classifications, or worse offer advice, based on superficial analysis of one's profile.

The most common models of traits incorporate four or five broad dimensions or factors. The least controversial dimension, observed as far back as the ancient Greeks, is simply extraversion vs. introversion (outgoing and physical–stimulation–oriented vs. quiet and physical–stimulation–averse).

Gordon Allport[27] delineated different kinds of traits, which he also called dispositions. Central traits are basic to an individual's personality, while secondary traits are more peripheral. Common traits are those recognized within a culture and thus may vary from culture to culture. Cardinal traits are those by which an individual may be strongly recognized.

Raymond Cattell's[28] research propagated a two–tiered personality structure with sixteen 'primary factors' (16 Personality Factors) and five 'secondary factors' (see Table 1.1). Cattell referred to these 16 factors as *primary factors*, as opposed to the so–called 'Big Five' factors which he considered *global factors*. All of the primary factors correlate with global factors and could therefore be considered subfactors within them.

[27] Gordon Willard Allport (November 11, 1897 - October 9, 1967) was an American psychologist. He was born in Montezuma, Indiana, the youngest of four brothers. One of his older brothers, Floyd Henry Allport, was an important and influential psychologist as well. Gordon W. Allport was a long time and influential member of the faculty at Harvard University from 1930-1967. His works include Becoming, Pattern and Growth in Personality, The Individual and his Religion, and perhaps his most influential book The Nature of Prejudice.

Allport was one of the first psychologists to focus on the study of the personality, and is often referred to as one of the fathers of personality psychology. Characteristically for this ecletic and pluralistic thinker, he was also an important contributor to social psychology as well. He rejected both a psychoanalytic approach to personality, which he thought often went too deep, and a behavioral approach, which he thought often did not go deep enough. He emphasized the uniqueness of each individual, and the importance of the present context, as opposed to past history, for understanding the personality.

[28] Raymond Bernard Cattell (20 March 1905 – 2 February 1998) was a British and American psychologist who theorized the existence of fluid and crystallized intelligences to explain human cognitive ability. He was famously productive throughout his 92 years, and ultimately was able to claim a combined authorship and co–authorship of 55 books and some 500 journal articles in addition to at least 30 standardized tests. His legacy includes not just that intellectual production, but also a spirit of scientific rigor brought to an otherwise soft science and kept burning by his students and co–researchers whom he was survived by.

In keeping with his devotion to rigorous scientific method, Cattell was an early proponent of the application in psychology of factor analytical methods, in place of what he called mere 'verbal theorizing.' One of the most important results of Cattell's application of factor analysis was the derivation of 16 factors underlying

Table 1.1. Cattell's 16 Personality Factors

Descriptors of Low Range	Primary Factor	Descriptors of High Range
Impersonal, distant, cool, reserved, detached, formal, aloof (Sizothymia)	Warmth	Warm, outgoing, attentive to others, kindly, easy going, participating, likes people (Affectothymia)
Concrete thinking, lower general mental capacity, less intelligent, unable to handle abstract problems (Lower Scholastic Mental Capacity)	Reasoning	Abstract–thinking, more intelligent, bright, higher general mental capacity, fast learner (Higher Scholastic Mental Capacity)
Reactive emotionally, changeable, affected by feelings, emotionally less stable, easily upset (Lower Ego Strength)	Emotional Stability	Emotionally stable, adaptive, mature, faces reality calm (Higher Ego Strength)
Deferential, cooperative, avoids conflict, submissive, humble, obedient, easily led, docile, accommodating (Submissiveness)	Dominance	Dominant, forceful, assertive, aggressive, competitive, stubborn, bossy (Dominance)
Serious, restrained, prudent, taciturn, introspective, silent (Desurgency)	Liveliness	Lively, animated, spontaneous, enthusiastic, happy go lucky, cheerful, expressive, impulsive (Surgency)
Expedient, nonconforming, disregards rules, self indulgent (Low Super Ego Strength)	Rule–Consciousness	Rule–conscious, dutiful, conscientious, conforming, moralistic, staid, rule bound (High Super Ego Strength)
Shy, threat–sensitive, timid, hesitant, intimidated (Threctia)	Social Boldness	Socially bold, venturesome, thick skinned, uninhibited (Parmia)
Utilitarian, objective, unsentimental, tough minded, self–reliant, no–nonsense, rough (Harria)	Sensitivity	Sensitive, aesthetic, sentimental, tender minded, intuitive, refined (Premsia)
Trusting, unsuspecting, accepting, unconditional, easy (Alaxia)	Vigilance	Vigilant, suspicious, skeptical, distrustful, oppositional (Protension)
Grounded, practical, prosaic, solution oriented, steady, conventional (Praxernia)	Abstractedness	Abstract, imaginative, absent minded, impractical, absorbed in ideas (Autia)
Forthright, genuine, artless, open, guileless, naive, unpretentious, involved (Artlessness)	Privateness	Private, discreet, nondisclosing, shrewd, polished, worldly, astute, diplomatic (Shrewdness)

Self–Assured, unworried, complacent, secure, free of guilt, confident, self satisfied (Untroubled)	Apprehension	Apprehensive, self doubting, worried, guilt prone, insecure, worrying, self blaming (Guilt Proneness)
Traditional, attached to familiar, conservative, respecting traditional ideas (Conservatism)	Openness to Change	Open to change, experimental, liberal, analytical, critical, free thinking, flexibility (Radicalism)
Group–oriented, affiliative, a joiner and follower dependent (Group Adherence)	Self–Reliance	Self–reliant, solitary, resourceful, individualistic, self sufficient (Self-Sufficiency)
Tolerated disorder, unexacting, flexible, undisciplined, lax, self-conflict, impulsive, careless of social rules, uncontrolled (Low Integration)	Perfectionism	Perfectionistic, organized, compulsive, self-disciplined, socially precise, exacting will power, control, self-sentimental (High Self–Concept Control)
Relaxed, placid, tranquil, torpid, patient, composed low drive (Low Ergic Tension)	Tension	Tense, high energy, impatient, driven, frustrated, over wrought, time driven. (High Ergic Tension)

A different model was proposed by Hans Eysenck,[29] who believed that just three traits: *extroversion*, *neuroticism* and *psychoticism* – were sufficient to describe human personality. Eysenck was one of the first psychologists to study personality with the method of *factor analysis*, a statistical technique introduced by Charles Spearman[30] and expanded by Raymond Cattell. Eysenck's

human personality. He called these 16 factors source traits because he believed that they provide the underlying source for the surface behaviors that we think of as personality. ('Psychology and Life, 7 ed.' by Richard Gerrig and Philip Zimbardo.) This theory of 16 personality factors and the instruments used to measure them are known collectively as the 16 Personality Factors.

[29] Hans Jürgen Eysenck (March 4, 1916 – September 4, 1997) was an eminent psychologist, most remembered for his work on intelligence and personality, though he worked in a wide range of areas. At the time of his death, Eysenck was the living psychologist most frequently cited in science journals.

Hans Eysenck was born in Germany, but moved to England as a young man in the 1930s because of his opposition to the Nazi party. Eysenck was the founding editor of the journal Personality and Individual Differences, and authored over 50 books and over 900 academic articles. He aroused intense debate with his controversial dealing with variation in IQ among racial groups.

[30] Charles Edward Spearman (September 10, 1863 - September 7, 1945) was an English psychologist known for work in statistics, as a pioneer of factor analysis, and for Spearman's rank correlation coefficient. He also did seminal work on models for human intelligence, including his theory that disparate cognitive test scores reflect a single general factor and coining the term g factor. Spearman had an unusual background for a psychologist. After 15 years as an officer in the British Army he resigned to study for a PhD in experimental psychology. In Britain

results suggested two main personality factors [Eys92a, Eys92b]. The first factor was the tendency to experience negative emotions, and Eysenck referred to it as 'neuroticism'. The second factor was the tendency enjoy positive events, especially social events, and Eysenck named it 'extraversion'. The two personality dimensions were described in his 1947 book 'Dimensions of Personality'. It is common practice in personality psychology to refer to the dimensions by the first letters, E and N. E and N provided a 2–dimensional space to describe individual differences in behavior. An analogy can be made to how latitude and longitude describe a point on the face of the earth. Also, Eysenck noted how these two dimensions were similar to the four personality types first proposed by the ancient Greek physician Galen[31]:

---

psychology was generally seen as a branch of philosophy and Spearman chose to study in Leipzig under Wilhelm Wundt. Besides Spearman had no conventional qualifications and Leipzig had liberal entrance requirements. He started in 1897 and after some interruption (he was recalled to the army during the South African War) he obtained his degree in 1906. He had already published his seminal paper on the factor analysis of intelligence (1904). Spearman met and impressed the psychologist William McDougall who arranged for Spearman to replace him when he left his position at University College London. Spearman stayed at University College until he retired in 1931. Initially he was Reader and head of the small psychological laboratory. In 1911 he was promoted to the Grote professorship of the Philosophy of Mind and Logic. His title changed to Professor of Psychology in 1928 when a separate Department of Psychology was created. When Spearman was elected to the Royal Society in 1924 the citation read "Dr. Spearman has made many researches in experimental psychology. His many published papers cover a wide field, but he is especially distinguished by his pioneer work in the application of mathematical methods to the analysis of the human mind, and his original studies of correlation in this sphere. He has inspired and directed research work by many pupils."

Spearman was strongly influenced by the work of Francis Galton. Galton did pioneering work in psychology and developed correlation, the main statistical tool used by Spearman. Spearman developed rank correlation (1904) and the widely used correction for attenuation (1907). His statistical work was not appreciated by his University College colleague Karl Pearson and there was long feud between them. Although Spearman achieved most recognition for his statistical work, he regarded this work as subordinate to his quest for the fundamental laws of psychology (see [WZZ03] for details).

[31] Galen, (Latin: Claudius Galenus of Pergamum) was an ancient Greek physician. The forename 'Claudius' is absent in Greek texts; it was first documented in texts from the Renaissance. Galen's views dominated European medicine for over a thousand years.

Galen transmitted Hippocratic medicine all the way to the Renaissance. His On the Elements According to Hippocrates describes the philosopher's system of four bodily humours, blood, yellow bile, black bile and phlegm, which were identified with the four classical elements, and in turn with the seasons. He created his own theories from those principles, and much of Galen's work can be seen as building on the Hippocratic theories of the body, rather than being purely innovative. In

1. High N and High E = Choleric type;
2. High N and Low E = Melancholic type;
3. Low N and High E = Sanguine type; and
4. Low N and Low E = Phlegmatic type.

The third dimension, 'psychoticism', was added to the model in the late 1970s, based upon collaborations between Eysenck and his wife, Sybil B.G. Eysenck, the current editor of Personality and Individual Differences (see [Eys69, Eys76]).

The major strength of Eysenck's model was to provide detailed theory of the causes of personality (see his 1985 book 'Decline and Fall of the Freudian Empire'). For example, Eysenck proposed that extraversion was caused by variability in cortical arousal; 'introverts are characterized by higher levels of activity than extraverts and so are chronically more cortically aroused than extraverts'. While it seems counterintuitive to suppose that introverts are more aroused than extraverts, the putative effect this has on behavior is such that the introvert seeks lower levels of stimulation. Conversely, the extravert seeks to heighten their arousal to a more optimal level (as predicted by the *Yerkes–Dodson Law*) by increased activity, social engagement and other stimulation–seeking behaviors.

Differences between Cattell and Eysenck emerged due to preferences for different forms of factor analysis, with Cattell using oblique, Eysenck orthogonal, rotation to analyze the factors that emerged when personality questionnaires were subject to statistical analysis. Today, the Big Five factors have the weight of a considerable amount of empirical research behind them. Building on the work of Cattell and others, Lewis Goldberg[32] proposed a five–dimensional personality model, nicknamed the 'Big Five' personality traits:

Extroversion (i.e., 'extroversion vs. introversion' above; outgoing and physical–stimulation–oriented vs. quiet and physical–stimulation–averse);

---

turn, he mainly ignored Latin writings of Celsus, but accepted that the ancient works of Asclepiades had sound theory.

Galen's own theories, in accord with Plato's, emphasized purposeful creation by a single Creator ('Nature' – Greek 'phusis') – a major reason why later Christian and Muslim scholars could accept his views. His fundamental principle of life was pneuma (air, breath) that later writers connected with the soul. These writings on philosophy were a product of Galen's well rounded education, and throughout his life Galen was keen to emphasize the philosophical element to medicine. Pneuma physicon (animal spirit) in the brain took care of movement, perception, and senses. Pneuma zoticon (vital spirit) in the heart controlled blood and body temperature. 'Natural spirit' in the liver handled nutrition and metabolism. However, he did not agree with the Pneumatist theory that air passed through the veins rather than blood.

[32] Lewis R. Goldberg is an American personality psychologist and a professor emeritus at the University of Oregon. Among his other accomplishments, Goldberg is closely associated with the Big Five taxonomy of personality. He has published well over 100 research articles and has been active on editorial boards.

1. Neuroticism (i.e., emotional stability; calm, unperturbable, optimistic vs. emotionally reactive, prone to negative emotions);
2. Agreeableness (i.e., affable, friendly, conciliatory vs. aggression aggressive, dominant, disagreeable);
3. Conscientiousness (i.e., dutiful, planful, and orderly vs. spontaneous, flexible, and unreliable); and
4. Openness to experience (i.e., open to new ideas and change vs. traditional and staid).

Character

A *character structure* is a system of relatively permanent motivational and other traits that are manifested in the characteristic ways that an individual relates to others and reacts to various kinds of challenges. The word 'structure' indicates that these several characteristics and/or learned patterns of behavior are linked in such a way as to produce a state that can be highly resistant to change. The idea has its roots in the work of Sigmund Freud[33] and several of his followers, the most important of whom (in this respect) is Erich Fromm.[34] Among other important participants in the establishment of this concept must surely be counted Erik Erikson.[35]

Among the earliest factors that determine an individual's eventual character structure are his or her genetic characteristics and early childhood nurture and education. A child who is well nurtured and taught in a relatively benign and consistent environment by loving adults who intend that the child should

[33] Sigmund Freud (May 6, 1856–September 23, 1939) was an Austrian neurologist and the founder of the psychoanalytic school of psychology. Freud is best known for his studies of sexual desire, repression, and the unconscious mind. He is commonly referred to as 'the father of psychoanalysis' and his work has been tremendously influential in the popular imagination–popularizing such notions as the unconscious, defence mechanisms, Freudian slips and dream symbolism – while also making a long-lasting impact on fields as diverse as literature, film, marxist and feminist theories, literary criticism, philosophy, and of course, psychology.

[34] Erich Pinchas Fromm (March 23, 1900 – March 18, 1980) was an internationally renowned German-American psychologist and humanistic philosopher. He is associated with what became known as the Frankfurt School of critical thinkers.

Central to Fromm's world view was his interpretation of the Talmud, which he began studying as a young man under Rabbi J. Horowitz and later studied under Rabbi Salman Baruch Rabinkow while working towards his doctorate in sociology at the University of Heidelberg and under Nehemia Nobel and Ludwig Krause while studying in Frankfurt. Fromm's grandfather and two great grandfathers on his father's side were rabbis, and a great uncle on his mother's side was a noted Talmudic scholar. However, Fromm turned away from orthodox Judaism in 1926 and turned towards secular interpretations of scriptural ideals.

[35] Erik Homburger Erikson (June 15, 1902 – May 12, 1994) was a developmental psychologist and psychoanalyst known for his theory on social development of human beings, and for coining the phrase identity crisis.

learn how to make objective appraisals regarding the environment will be likely to form a normal or productive character structure. On the other hand, a child whose nurture and/or education are not ideal, living in a treacherous environment and interacting with adults who do not take the long–term interests of the child to heart will be more likely to form a pattern of behavior that suits the child to avoid the challenges put forth by a malign social environment. The means that the child invents to make the best of a hostile environment. Although this may serve the child well while in that bad environment, it may also cause the child to react in inappropriate ways, ways damaging to his or her own interests, when interacting with people in a more ideal social context. Major trauma that occurs later in life, even in adulthood, can sometimes have a profound effect. However, character may also develop in a positive way according to how the individual meets the psychosocial challenges of the life cycle (Erikson).

Freud's first paper on character described the anal character consisting of stubbornness, stinginess and extreme neatness. He saw this as a reaction formation to the child's having to give up pleasure in anal eroticism.The positive version of this character is the conscientious, inner directed obsessive. Freud also described the erotic character as both loving and dependent. And the narcissistic character as the natural leader, aggressive and independent because of not internalizing a strong super ego.

For Erich Fromm, character develops as the way in which an individual structures modes of assimilation and relatedness. The character types are almost identical to Freud's but Fromm gives them different names, receptive, hoarding, exploitative. Fromm adds the marketing type as the person who continually adapts the self to succeed in the new service economy. For Fromm, character types can be productive or unproductive. Fromm notes that character structures develop in each individual to enable him or her to interact successfully within a given society, to adapt to its mode of production and social norms may be very counter–productive when used in a different society.

Wisdom

On the other hand, *wisdom* is the ability, developed through experience, insight and reflection, to discern truth and exercise good judgment. It is sometimes conceptualized as an especially well developed form of common sense. Most psychologists regard wisdom as distinct from the cognitive abilities measured by standardized intelligence tests. Wisdom is often considered to be a trait that can be developed by experience, but not taught. When applied to practical matters, the term wisdom is synonymous with prudence. Some see wisdom as a quality that even a child, otherwise immature, may possess independent of experience or complete knowledge. The status of wisdom or prudence as a virtue is recognized in cultural, philosophical and religious sources. Some define wisdom in a utilitarian sense, as foreseeing consequences and acting to maximize the long–term common good.

A standard philosophical definition says that wisdom consists of making the best use of available knowledge. As with all decisions, a wise decision may be made with incomplete information. The technical philosophical term for the opposite of wisdom is folly. For example, in his *Metaphysics*, Aristotle defines wisdom as knowledge of causes: why things exist in a particular fashion.

Beyond the simple expedient of experience (which may be considered the most difficult way to gain wisdom as through the 'school of hard knocks'), there are a variety of other avenues to gaining wisdom which vary according to different philosophies. For example, the so–called *freethinkers*[36]believe that wisdom may come from pure reason and perhaps experience. Recall that *freethought* is a philosophical doctrine that holds that beliefs should be formed on the basis of science and logical principles and not be comprised by authority, tradition or any other dogmatic or otherwise fallacious belief system that restricts logical reasoning. The cognitive application of freethought is known as *freethinking*, and practitioners of freethought are known as freethinkers. Freethought holds that individuals should neither accept nor reject ideas proposed as truth without recourse to knowledge and reason. Thus, freethinkers strive to build their beliefs on the basis of facts, scientific inquiry, and logical principles, independent of the factual/logical fallacies and intellectually-limiting effects of authority, cognitive bias, conventional wisdom, popular culture, prejudice, sectarianism, tradition, urban legend and all other dogmatic or otherwise fallacious principles. When applied to religion, the philosophy of freethought holds that, given presently–known facts, established scientific theories, and logical principles, there is insufficient evidence to support the existence of supernatural phenomena. A line from 'Clifford's Credo' by the 19th Century British mathematician and philosopher William Clifford[37] perhaps best describes the premise of freethought: "It is wrong always,

[36] Freethought is a philosophical doctrine that holds that beliefs should be formed on the basis of science and logical principles and not be comprised by authority, tradition or any other dogmatic or otherwise fallacious belief system that restricts logical reasoning. The cognitive application of freethought is known as freethinking, and practitioners of freethought are known as freethinkers.

[37] William Kingdon Clifford, FRS (May 4, 1845 – March 3, 1879) was an English mathematician who also wrote a fair bit on philosophy. Along with Hermann Grassmann, he invented what is now termed *geometric algebra*, a special case being the *Clifford algebras* named in his honour, which play a role in contemporary mathematical physics. He was the first to suggest that gravitation might be a manifestation of an underlying geometry. His philosophical writings coined the phrase 'mind–stuff'.

Influenced by Riemann and Lobachevsky, Clifford studied non–Euclidean geometry. In 1870, he wrote On the space theory of matter, arguing that energy and matter are simply different types of curvature of space. These ideas later played a fundamental role in Albert Einstein's general theory of relativity. Yet Clifford is now best remembered for his eponymous Clifford algebras, a type of associative algebra that generalizes the complex numbers and William Rowan Hamilton's *quaternions*. The latter resulted in the octonions (biquaternions),

everywhere, and for anyone, to believe anything upon insufficient evidence." Since many popular beliefs are based on dogmas, freethinkers' opinions are often at odds with commonly–established views.

On the other hand, there is also a common belief that wisdom comes from *intuition* or, 'superlogic', as it is called by Tony Buzan,[38] inventor of *mind maps*. For example, *holists* believe that wise people sense, work with and align themselves and others to life. In this view, wise people help others appreciate the fundamental interconnectedness of life. Also, some religions hold that wisdom may be given as a gift from God. For example, *Buddha* taught that a wise person is endowed with good bodily conduct, good verbal conduct and good mental conduct and a wise person does actions that are unpleasant to do but give good results and doesn't do actions that are pleasant to do but give bad results; this is called *karma*. According to *Hindu scriptures*, spiritual wisdom – *jnana* alone can lead to liberation. *Confucius* stated that wisdom can be learned by three methods: (i) *reflection* (the noblest), (ii) *imitation* (the easiest) and (iii) *experience* (the bitterest).

1.1.1 Human Intelligence

At least two major 'consensus' definitions of intelligence have been proposed. First, from 'Intelligence: Knowns and Unknowns', a report of a task force convened by the *American Psychological Association*[39] in 1995 (see [APS98]):

---

which he employed to study motion in non–Euclidean spaces and on certain surfaces, now known as Klein–Clifford spaces. He showed that spaces of constant curvature could differ in topological structure. He also proved that a Riemann surface is topologically equivalent to a box with holes in it. On Clifford algebras, quaternions, and their role in contemporary mathematical physics.

[38] Tony Buzan (1942–) is the originator of mind mapping and coined the term mental literacy. He was born in London and received double Honours in psychology, English, mathematics and the General Sciences from the University of British Columbia in 1964. He is probably best known for his book, Use Your Head, his promotion of mnemonic systems and his mind–mapping techniques. Following his 1970s series for the BBC, many of his ideas have been set into his series of five books: Use Your Memory, Master Your Memory, Use Your Head, The Speed Reading Book and The Mind Map Book.

In essence, Buzan teaches "Learn how your brain learns rapidly and naturally." His work is partly based on the explosion of brain research that has taken place since the late 1950s, and the work on the left and right brain by psychologist Robert Ornstein and Nobel Laureate Roger Wolcott Sperry.

[39] The American Psychological Association (APA) is a professional organization representing psychology in the US. It has around 150,000 members and an annual budget of around $70m. The APA mission statement is to "advance psychology as a science and profession and as a means of promoting health, education, and human welfare." The APA was founded in July 1892 at Clark University by a group of 26 men. Its first president was G. Stanley Hall. There are currently 54

Individuals differ from one another in their ability to understand complex ideas, to adapt effectively to the environment, to learn from experience, to engage in various forms of reasoning, to overcome obstacles by taking thought. Although these individual differences can be substantial, they are never entirely consistent: a given person's intellectual performance will vary on different occasions, in different domains, as judged by different criteria. Concepts of 'intelligence' are attempts to clarify and organize this complex set of phenomena.

A second definition of intelligence comes from the 'Mainstream Science on Intelligence', which was signed by 52 intelligence researchers in 1994 (also see [APS98]): Intelligence is a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience. It is not merely book learning, a narrow academic skill, or test–taking smarts. Rather, it reflects a broader and deeper capability for comprehending our surroundings, i.e., 'catching on', 'making sense' of things, or 'figuring out' what to do.

Individual intelligence experts have offered a number of similar definitions:

(i) David Wechsler:[40] " ... the aggregate or global capacity of the individual to act purposefully, to think rationally, and to deal effectively with his environment."
(ii) Cyril Burt:[41] " ... innate general cognitive ability."
(iii) Howard Gardner:[42] " To my mind, a human intellectual competence must entail a set of skills of problem solving, enabling the individual to resolve genuine problems or difficulties that he or she encounters and, when appropriate, to create an effective product, and must also entail the potential for finding or creating problems, and thereby laying the groundwork for the acquisition of new knowledge."

---

professional divisions in the APA. It is affiliated with 58 state and territorial and Canadian provincial associations.

[40] David Wechsler (January 12, 1896, Lespedi, Romania – May 2, 1981, New York, New York) was a leading Romanian-American psychologist. He developed well–known intelligence scales, such as the Wechsler Adult Intelligence Scale (WAIS) and the Wechsler Intelligence Scale for Children (WISC).

[41] Sir Cyril Lodowic Burt (March 3, 1883 — October 10, 1971) was a prominent British educational psychologist. He was a member of the London School of Differential Psychology. Some of his work was controversial for its conclusions that genetics substantially influence mental and behavioral traits. After his death, he was famously accused of scientific fraud.

[42] Howard Gardner (born in Scranton, Pennsylvania, USA in 1943) is a psychologist based at Harvard University best known for his theory of multiple intelligences. In 1981 he was awarded a MacArthur Prize Fellowship.

(iv) Richard Herrnstein[43] and Charles Murray: " ... cognitive ability."
(v) Robert Sternberg:[44] "... goal–directed adaptive behavior."

Psychometric Definition of Intelligence and Its Criticisms

Despite the variety of concepts of intelligence, the most influential approach to understanding intelligence (i.e., with the most supporters and the most published research over the longest period of time) is based on *psychometric testing*,[45] which regards intelligence as *cognitive ability*.

[43] Richard J. Herrnstein (May 20, 1930 – September 13, 1994) was a prominent researcher in comparative psychology who did pioneering work on pigeon intelligence employing the Experimental Analysis of Behavior and formulated the 'Matching Law' in the 1960s, a breakthrough in understanding how reinforcement and behavior are linked. He was the Edgar Pierce Professor of psychology at Harvard University and worked with B. F. Skinner in the Harvard pigeon lab, where he did research on choice and other topics in behavioral psychology. Herrnstein became more broadly known for his work on the correlation between race and intelligence, first in the 1970s, then with Charles Murray, discussed in their controversial best–selling 1994 book, The Bell Curve. Herrnstein described the behavior of hyperbolic discounting, in which people will choose smaller payoffs sooner instead of larger payoffs later. He developed a type of non–parametric statistics that he dubbed ρ.

[44] Robert J. Sternberg (borne 8 December 1949) is a psychologist and psychometrician and the Dean of Arts and Sciences at Tufts University. He was formerly IBM Professor of Psychology and Education at Yale University and the President of the American Psychological Association. Sternberg currently sits on the editorial board of Intelligence. Sternberg has proposed the so–called *Triarchic theory of intelligence* and a triangular theory of love. He is the creator (with Todd Lubart) of the investment theory of creativity, which states that creative people buy low and sell high in the world of ideas, and a propulsion theory of creative contributions, which states that creativity is a form of leadership.

[45] Psychometrics is the field of study concerned with the theory and technique of psychological measurement, which includes the measurement of knowledge, abilities, attitudes, and personality traits. The field is primarily concerned with the study of differences between individuals. It involves two major research tasks, namely: (i) the construction of instruments and procedures for measurement; and (ii) the development and refinement of theoretical approaches to measurement. Much of the early theoretical and applied work in psychometrics was undertaken in an attempt to measure intelligence. The origin of psychometrics has connections to the related field of psychophysics. Charles Spearman, a pioneer in psychometrics who developed approaches to the measurement of intelligence, studied under Wilhelm Wundt and was trained in psychophysics. The psychometrician L.L. Thurstone later developed and applied a theoretical approach to the measurement referred to as the law of comparative judgment, an approach which has close connections to the psychophysical theory developed by Ernst Heinrich Weber and Gustav Fechner. In addition, Spearman and Thurstone both made important contributions to the theory and application of factor analysis, a statistical

Recall that *psychometrics* is the field of study concerned with the theory and technique of psychological measurement, which includes the measurement of knowledge, abilities, attitudes, and personality traits. The field is primarily concerned with the study of differences between individuals. It involves two major research tasks, namely:

(i) the construction of instruments and procedures for measurement; and

(ii) the development and refinement of theoretical approaches to measurement. Much of the early theoretical and applied work in psychometrics was undertaken in an attempt to measure intelligence.

The origin of psychometrics has connections to the related field of psychophysics. Charles Spearman, a pioneer in psychometrics who developed approaches to the measurement of intelligence, studied under Wilhelm Wundt[46] and was trained in psychophysics. The psychometrician Louis

---

method that has been used extensively in psychometrics. More recently, psychometric theory has been applied in the measurement of personality, attitudes and beliefs, academic achievement, and in health-related fields. Measurement of these unobservable phenomena is difficult, and much of the research and accumulated art in this discipline has been developed in an attempt to properly define and quantify such phenomena. Critics, including practitioners in the physical sciences and social activists, have argued that such definition and quantification is impossibly difficult, and that such measurements are often misused. Proponents of psychometric techniques can reply, though, that their critics often misuse data by not applying psychometric criteria, and also that various quantitative phenomena in the physical sciences, such as heat and forces, cannot be observed directly but must be inferred from their manifestations. Figures who made significant contributions to psychometrics include Karl Pearson, L. L. Thurstone, Georg Rasch and Arthur Jensen.

[46] Wilhelm Maximilian Wundt (August 16, 1832–August 31, 1920) was a German physiologist and psychologist. He is generally acknowledged as a founder of experimental psychology and cognitive psychology. He is less commonly recognised as a founding figure in social psychology, however, the later years of Wundt's life were spent working on Völkerpsychologie which he understood as a study into the social basis of higher mental functioning.

Wundt combined philosophical introspection with techniques and laboratory apparatuses brought over from his physiological studies with Helmholtz, as well as many of his own design. This experimental introspection was in contrast to what had been called psychology until then, a branch of philosophy where people introspected themselves. Wundt argued in his 1904 book 'Principles of Physiological Psychology' that "we learn little about our minds from casual, haphazard self–observation ... It is essential that observations be made by trained observers under carefully specified conditions for the purpose of answering a well–defined question."

The methods Wundt used are still used in modern psychophysical work, where reactions to systematic presentations of well–defined external stimuli are measured in some way–reaction time, reactions, comparison with graded colors or sounds, and so forth. His chief method of investigation was called *introspection* in the terminology of the time, though *observation* may be a better translation.

Thurstone[47] later developed and applied a theoretical approach to the measurement referred to as the law of comparative judgment, an approach which has close connections to the psychophysical theory developed by Ernst Weber and Gustav Fechner (see below). In addition, Spearman and Thurstone both made important contributions to the theory and application of factor analysis, a statistical method that has been used extensively in psychometrics. More recently, psychometric theory has been applied in the measurement of personality, attitudes and beliefs, academic achievement, and in health–related fields. Measurement of these unobservable phenomena is difficult, and much of the research and accumulated art in this discipline has been developed in an attempt to properly define and quantify such phenomena. Critics, including practitioners in the physical sciences and social activists, have argued that such definition and quantification is impossibly difficult, and that such measurements are often misused. Proponents of psychometric techniques can reply, though, that their critics often misuse data by not applying psychometric criteria, and also that various quantitative phenomena in the physical sciences, such as heat and forces, cannot be observed directly but must be inferred from their manifestations. Figures who made significant contributions to psychometrics include Karl Pearson, Louis Thurstone, Georg Rasch and Arthur Jensen.

Wundt subscribed to a 'psychophysical parallelism' (which entirely excludes the possibility of a mind–body/cause–effect relationship), which was supposed to stand above both materialism and idealism. His epistemology was an eclectic mixture of the ideas of Spinoza, Leibniz, Kant, and Hegel.

[47] Louis Leon Thurstone (29 May 1887–29 September 1955) was a U.S. pioneer in the fields of psychometrics and psychophysics. He conceived the approach to measurement known as the law of comparative judgment, and is well known for his contributions to *factor analysis*. He is responsible for the standardized mean and standard deviation of IQ scores used today, as opposed to the Intelligence Test system originally used by Alfred Binet. He is also known for the development of the Thurstone scale.

Thurstone's work in factor analysis led him to formulate a model of intelligence center around 'Primary Mental Abilities' (PMAs), which were independent group factors of intelligence that different individuals possessed in varying degrees. He opposed the notion of a singular general intelligence that factored into the scores of all psychometric tests and was expressed as a mental age. This idea was unpopular at the time due to its obvious conflicts with Spearman's 'mental energy' model, and is today still largely discredited. Nonetheless, Thurstone's contributions to methods of factor analysis have proved invaluable in establishing and verifying later psychometric factor structures, and has influenced the hierarchical models of intelligence in use in intelligence tests such as WAIS and the modern Stanford–Binet IQ test.

The *seven primary mental abilities* in Thurstone's model were *verbal comprehension, word fluency, number facility, spatial visualization, associative memory, perceptual speed and reasoning.*

Intelligence, narrowly defined by psychometrics, can be measured by intelligence tests, also called *intelligence quotient* (IQ)[48] tests. Such intelligence tests take many forms, but the common tests (*Stanford–Binet*,[49] *Raven's Progressive Matrices*,[50] Wechsler Adult Intelligence

[48] An intelligence quotient or IQ is a score derived from a set of standardized tests of intelligence. Intelligence tests come in many forms, and some tests use a single type of item or question. Most tests yield both an overall score and individual sub–tests scores. Regardless of design, all IQ tests measure the same general intelligence. Component tests are generally designed and chosen because they are found to be predictable of later intellectual development, such as educational achievement. IQ also correlates with job performance, socioeconomic advancement, and 'social pathologies'. Recent work has demonstrated links between IQ and health, longevity, and functional literacy. However, IQ tests do not measure all meanings of 'intelligence', such as creativity. IQ scores are relative (like placement in a race), not absolute (like the measurement of a ruler). The average IQ scores for many populations were rising during the 20th century: a phenomenon called the *Flynn effect*. It is not known whether these changes in scores reflect real changes in intellectual abilities. On average, IQ scores are stable over a person's lifetime, but some individuals undergo large changes. For example, scores can be affected by the presence of learning disabilities.

[49] The modern field of intelligence testing began with the Stanford-Binet IQ test. The Stanford-Binet itself started with the French psychologist Alfred Binet who was charged by the French government with developing a method of identifying intellectually deficient children for placement in special education programs. As Binet indicated, case studies may be more detailed and at times more helpful, but the time required to test large numbers of people would be huge. Unfortunately, the tests he and his assistant Victor Henri developed in 1896 were largely disappointing [Fan85].

[50] Raven's Progressive Matrices are widely used non–verbal intelligence tests. In each test item, one is asked to find the missing part required to complete a pattern. Each Set of items gets progressively harder, requiring greater cognitive capacity to encode and analyze. The test is considered by many intelligence experts to be one of the most **g**–loaded in existence. The matrices are offered in three different forms for different ability levels, and for age ranges from five through adult: (i) Colored Progressive Matrices (younger children and special groups); (ii) Standard Progressive Matrices (average 6 to 80 year olds); and (iii) Advanced Progressive Matrices (above average adolescents and adults). According to their author, Raven's Progressive Matrices and Vocabulary tests measure the two main components of general intelligence (originally identified by Spearman): the ability to think clearly and make sense of complexity, which is known as eductive ability (from the Latin root 'educere', meaning 'to draw out'; and the ability to store and reproduce information, known as reproductive ability. Adequate standardization, ease of use (without written or complex instructions), and minimal cost per person tested are the main reasons for its widespread international use in most countries of the world. It appears to measure a type of *reasoning ability* which is fundamental to making sense out of the 'booming buzzing confusion' in

Scale,[51] *Wechsler–Bellevue I*,[52] and others) all measure the same dominant form of intelligence, **g** or 'general intelligence factor'. The abstraction of **g** stems from the observation that scores on all forms of cognitive tests *positively correlate* with one another. **g** can be derived as the *principal intelligence factor* from *cognitive test scores* using the *multivariate correlation statistical method* of *factor analysis* (FA).

---

all walks of life. Thus, it has among the highest predictive validities of any test in most occupational groups and, even more importantly, in predicting social mobility ... the level of job a person will attain and retain. Although it is sometimes criticized for being costly, this is based on a failure to calculate cost per person tested with re–usable test booklets that can be used up to 50 times each. The authors of the Manual recommend that, when used in selection, RPM scores are set in the context of information relating to Raven's framework for the assessment of Competence. Some of the most fundamental research in cognitive psychology has been carried out with the RPM. The tests have been shown to work–scale–measure the same thing – in a vast variety of cultural groups. There is no truth in the assertion that the low mean scores obtained in some groups arise from a general lack of familiarity with the way of thought measured by the test. Two remarkable, and relatively recent, findings are that, on the one hand, the actual scores obtained by people living in most countries with a tradition of literacy – from China, Russia, and India through Europe to Kuwait – are very similar at any point in time. On the other hand, in all countries, the scores have increased dramatically over time ... such that 50% of our grandparents would be assigned to special education classes if they were judged against today's norms. Yet none of the common explanations (e.g., access to television, changes in education, changes in family size etc.) hold up. The explanation seems to have more in common with those put forward to explain the parallel increase in life expectancy ... which has doubled over the same period of time.

[51] Wechsler Adult Intelligence Scale or WAIS is a general IQ test, published in February 1955 as a revision of the Wechsler–Bellevue test (1939), standardized for use with adults over the age of 16. In this test intelligence is quantified as the global capacity of the individual to act purposefully, to think rationally, and to deal effectively with the environment.

[52] David Wechsler (January 12, 1896, Lespedi, Romania – May 2, 1981, New York, New York) was a leading Romanian–American psychologist. He developed well–known intelligence scales, such as the Wechsler Adult Intelligence Scale (WAIS) and the Wechsler Intelligence Scale for Children (WISC). The Wechsler Adult Intelligence Scale (WAIS) was developed first in 1939 and then called the Wechsler–Bellevue Intelligence Test. From these he derived the Wechsler Intelligence Scale for Children (WISC) in 1949 and the Wechsler Preschool and Primary Scale of Intelligence (WPPSI) in 1967. Wechsler originally created these tests to find out more about his patients at the Bellevue clinic and he found the then–current *Binet IQ test* unsatisfactory. The tests are still based on his philosophy that intelligence is "the global capacity to act purposefully, to think rationally, and to deal effectively with (one's) environment."

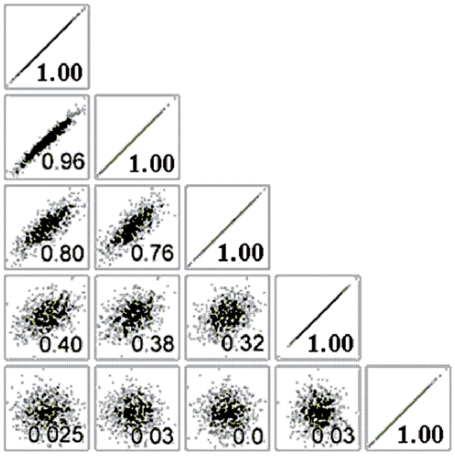

Fig. 1.1. Example of positive linear correlations between 1000 pairs of numbers. Note that each set of points correlates maximally with itself, as shown on the diagonal. Also, note that we have not plot the upper part of the correlation matrix as it is symmetrical.

Correlation and Factor Analysis

Recall that *correlation*, also called *correlation coefficient*, indicates the strength and direction of a linear relationship between two random variables (see Figure 1.1). In other words, correlation is a measure of the relation between two or more statistical variables. In general statistical usage, correlation (or, co–rrelation) refers to the departure of two variables from independence, although correlation does not imply their *functional causal relation*. In this broad sense there are several coefficients, measuring the degree of correlation, adapted to the nature of data. A number of different coefficients are used for different situations. Correlation coefficients can range from -1.00 to $+1.00$. The value of -1.00 represents a perfect negative correlation while a value of $+1.00$ represents a perfect positive correlation. The perfect correlation indicates an existence of functional relation between two statistical variables. A value of 0.00 represents a lack of correlation.Geometrically, the correlation coefficient can also be viewed as the cosine of the angle between the two vectors of samples drawn from the two random variables.

The most widely–used type of correlation simple linear coefficient is Pearson r, also called *linear* or *product–moment* correlation, which assumes that the two variables are measured on at least interval scales, and it determines the extent to which values of the two variables are 'proportional' to each other. The value of correlation coefficient does not depend on the specific measurement units used. Proportional means linearly related using regression line or least squares line. If the correlation coefficient is squared, then the resulting value (r^2, the *coefficient of determination*) will represent the proportion of

common variation in the two variables (i.e., the 'strength' or 'magnitude' of the relationship). In order to evaluate the correlation between variables, it is important to know this 'magnitude' or 'strength' as well as the significance of the correlation.

The significance level calculated for each correlation is a primary source of information about the reliability of the correlation. The significance of correlation coefficient of particular magnitude will change depending on the size of the sample from which it was computed. The test of significance is based on the assumption that each of the two variables is normally distributed and that their bivariate ('combined') distribution is normal (which can be tested by examining the 3D bivariate distribution histogram). However, *Monte–Carlo* studies suggest that meeting those assumptions (especially the second one) is not absolutely crucial if our sample size is not very small and when the departure from normality is not very large. It is impossible to formulate precise recommendations based on those Monte–Carlo results, but many researchers follow a rule of thumb that if our sample size is 50 or more then serious biases are unlikely, and if our sample size is over 100 then you should not be concerned at all with the normality assumptions.

Recall that the *normal distribution*, also called *Gaussian distribution*, is an extremely important probability distribution in many fields. It is a family of distributions of the same general form, differing in their location and scale parameters: the *mean* ('average') μ and *standard deviation* ('variability') σ, respectively. The *standard normal distribution* is the normal distribution with a mean of zero and a standard deviation of one. It is often called the *bell curve* because the graph of its *probability density function pdf*, given by the *Gaussian function*

$$pdf = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

resembles a bell shape (here, $\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-x^2/2}$ is the *pdf* for the standard normal distribution). The corresponding *cumulative distribution function cdf* is defined as the probability that a variable X has a value less than or equal to x, and it is expressed in terms of the *pdf* as

$$cdf = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{(u-\mu)^2}{2\sigma^2}\right) du.$$

Now, the correlation $r_{X,Y}$ between two *normally distributed random variables* X and Y with expected values μX and μY and standard deviations σX and σY is defined as:

$$r_{XY} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y},$$

where E denotes the expected value of the variable and cov means covariance. Since $\mu X = E(X)$, $\sigma_X^2 = E(X^2) - E^2(X)$ and similarly for Y, we can write (see, e.g., [CCW03])

$$r_{XY} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\,\sqrt{E(Y^2) - E^2(Y)}}.$$

Assume that we have a data matrix $\mathbf{X} = \{x_{i\alpha}\}$ formed out of the *sample* $\{\mathbf{x_i}\}$ of n normally distributed simulator tests called observable–vectors or *manifest variables*, defined on the sample $\{\alpha = 1, \ldots, N\}$ of pilot (for the statistical significance the practical user's criterion is $N \geq 5n$). The *maximum likelihood estimator* of the Pearson correlation coefficient r_{ik} between any two manifest variables $\mathbf{x_i}$ and $\mathbf{x_k}$ is defined as[53]

$$r_{ik} = \frac{\sum_{\alpha=1}^{N}(x_{i\alpha} - \boldsymbol{\mu}_i)(x_{k\alpha} - \boldsymbol{\mu}_k)}{\sqrt{\sum_{\alpha=1}^{N}(x_{i\alpha} - \boldsymbol{\mu}_i)^2}\sqrt{\sum_{\alpha=1}^{N}(x_{k\alpha} - \boldsymbol{\mu}_k)^2}},$$

where

$$\mu_i = \frac{1}{N}\sum_{\alpha=1}^{N} x_{i\alpha}$$

is the arithmetic mean of the variable $\mathbf{x_i}$.[54] Correlation matrix $\mathbf{R}$ is the matrix $\mathbf{R} \equiv \mathbf{R}_{ik} = \{r_{ik}\}$ including $n \times n$ Pearson correlation coefficients r_{ik} calculated between n manifest variables $\{\mathbf{x_i}\}$. Therefore, $\mathbf{R}$ is symmetrical matrix

[53] A time–dependent generalization $C_{\alpha\beta} = C_{\alpha\beta}(t)$ of the correlation coefficient r_{XY} is the *correlation function*, defined as follows. For the two time–series, $x_\alpha(t_i)$ and $x_\beta(t_i)$ of the same length $(i = 1, \ldots, T)$, one defines the correlation function by

$$C_{\alpha\beta} = \frac{\sum_i (x_\alpha(t_i) - \bar{x}_\alpha)(x_\beta(t_i) - \bar{x}_\beta)}{\sqrt{\sum_i (x_\alpha(t_i) - \bar{x}_\alpha)^2 \sum_j (x_\beta(t_j) - \bar{x}_\beta)^2}},$$

where $\bar{x}$ denotes a time average over the period studied. For two sets of N time–series $x_\alpha(t_i)$ each $(\alpha, \beta = 1, \ldots, N)$ all combinations of the elements $C_{\alpha\beta}$ can be used as entries of the $N \times N$ correlation matrix $\mathbf{C}$. By diagonalizing $\mathbf{C}$, i.e., solving the eigenvalue problem:

$$\mathbf{C}\mathbf{v}^k = \lambda_k \mathbf{v}^k,$$

one gets the eigenvalues λ_k $(k = 1, \ldots, N)$ and the corresponding eigenvectors $\mathbf{v}^k = \{v_\alpha^k\}$.

[54] The following algorithm (in pseudocode) estimates bivariate correlation coefficient with good numerical stability:

```
Begin
  sum_sq_x = 0;
  sum_sq_y = 0;
  sum_coproduct = 0;
  mean_x = x[1];
  mean_y = y[1];
```

with ones on the main diagonal. The correlation matrix **R** represents the total variability of all included manifest variables. In other words it stores all information about all simulator tests and all pilot. Now, if the number of included simulator tests is small, this information is meaningful for the human mind. But if we perform one hundred tests (on five hundred pilot), then the correlation matrix contains ten thousand Pearson correlation coefficients. This is the reason for seeking the 'latent' factor structure, underlying the whole co–variability contained in the correlation matrix.

Therefore, the correlation is defined only if both of the standard deviations are finite and both of them are nonzero. It is a corollary of the *Cauchy–Schwarz inequality*[55] that the correlation cannot exceed 1 in absolute value.

```
    for i in 2 to N:
      sweep = (i - 1.0) / i;
      delta_x = x[i] - mean_x;
      delta_y = y[i] - mean_y;
      sum_sq_x += delta_x * delta_x * sweep;
      sum_sq_y += delta_y * delta_y * sweep;
      sum_coproduct += delta_x * delta_y * sweep;
      mean_x += delta_x / i;
      mean_y += delta_y / i ;
    end_for;
    pop_sd_x = sqrt( sum_sq_x / N );
    pop_sd_y = sqrt( sum_sq_y / N );
    cov_x_y = sum_coproduct / N;
    correlation = cov_x_y / (pop_sd_x * pop_sd_y);
  End.
```

[55] The Cauchy–Schwarz inequality, named after Augustin Louis Cauchy (the father of complex analysis) and Hermann Amandus Schwarz, is a useful inequality encountered in many different settings, such as linear algebra applied to vectors, in analysis applied to infinite series and integration of products, and in probability theory, applied to variances and covariances. The Cauchy–Schwarz inequality states that if x and y are elements of real or complex inner product spaces then

$$|\langle x, y\rangle|^2 \leq \langle x, x\rangle \cdot \langle y, y\rangle.$$

The two sides are equal iff x and y are linearly dependent (or in geometrical sense they are parallel). This contrasts with a property that the inner product of two vectors is zero if they are orthogonal (or perpendicular) to each other. The inequality hence confers the notion of *the angle between the two vectors* to an inner product, where concepts of *Euclidean geometry* may not have meaningful sense, and justifies that the notion that inner product spaces are generalizations of *Euclidean space*.

An important consequence of the Cauchy–Schwarz inequality is that the inner product is a continuous function.

Another form of the Cauchy–Schwarz inequality is given using the notation of norm, as explained under norms on inner product spaces, as

The correlation is 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, and some value in between in all other cases, indicating the degree of linear dependence between the variables. The closer the coefficient is to either -1 or 1, the stronger the correlation between the variables (see Figure 1.1). If the variables are independent then the correlation is 0, but the converse is not true because the correlation coefficient detects only linear dependencies between two variables. For example, suppose the random variable X is uniformly distributed on the interval from -1 to 1, and $Y = X^2$. Then Y is completely determined by X, so that X and Y are dependent, but their correlation is zero; this means that they are uncorrelated. The correlation matrix of n random variables $X_1, \ldots, X_n$ is the $n \times n$ matrix whose ij entry is $r_{X_i X_j}$. If the measures of correlation used are product–moment coefficients, the correlation matrix is the same as the covariance matrix of the standardized random variables X_i/σ_{X_i} (for $i = 1, \ldots, n$). Consequently it is necessarily a non–negative definite matrix. The correlation matrix is symmetrical (the correlation between X_i and X_j is the same as the correlation between X_j and X_i).

As a higher derivation of the correlation matrix analysis and its eigenvectors, the so–called principal components, the *factor analysis* (FA) is a multivariate statistical technique used to explain variability among a large set of observed random variables in terms of fewer unobserved random 'latent' variables, called *factors*. The observed, or 'manifested' variables are modelled as linear combinations of the factors, plus 'error terms'. According to FA, classical bivariate correlation analysis is an artificial extraction from a rial multivariate world, especially in human sciences. FA originated in psychometrics, and is used in social sciences, marketing, product management, operations research, and other applied sciences that deal with large multivariate quantities of data.

For example,[56] suppose a psychologist proposes a theory that there are two kinds of intelligence, 'verbal intelligence' and 'mathematical intelligence'. Note that these are inherently unobservable. Evidence for the theory is sought in the examination scores of 1000 students in each of 10 different academic fields. If a student is chosen randomly from a large population, then the student's 10 scores are random variables. The psychologist's theory may say that the average score in each of the 10 subjects for students with a particular level of verbal intelligence and a particular level of mathematical intelligence is a certain number times the level of verbal intelligence plus a certain number times the level of mathematical intelligence, i.e., it is a linear combination of those two 'factors'. The numbers by which the two 'intelligences' are multiplied

---

$$|\langle x, y\rangle| \leq \|x\| \cdot \|y\|.$$

[56] This oversimplified example should not be taken to be realistic. Usually we are dealing with many factors.

are posited by the theory to be the same for all students, and are called 'factor loadings'. For example, the theory may hold that the average student's aptitude in the field of amphibology is

{ 10 × the student's verbal intelligence } + { 6 × the student's mathematical intelligence }.

The numbers 10 and 6 are the factor loadings associated with amphibology. Other academic subjects may have different factor loadings. Two students having identical degrees of verbal intelligence and identical degrees of mathematical intelligence may have different aptitudes in amphibology because individual aptitudes differ from average aptitudes. That difference is called the 'error' — an unfortunate misnomer in statistics that means the amount by which an individual differs from what is average. The observable data that go into factor analysis would be 10 scores of each of the 1000 students, a total of 10,000 numbers. The factor loadings and levels of the two kinds of intelligence of each student must be inferred from the data. Even the number of factors (two, in this example) must be inferred from the data.

In the example above, for $i = 1, \ldots, 1,000$ the ith student's scores are

$$\begin{array}{ccccccc} x_{1,i} & = & \mu_1 & + & \ell_{1,1} v_i & + & \ell_{1,2} m_i & + & \varepsilon_{1,i} \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ x_{10,i} & = & \mu_{10} & + & \ell_{10,1} v_i & + & \ell_{10,2} m_i & + & \varepsilon_{10,i} \end{array}$$

where $x_{k,i}$ is the ith student's score for the kth subject, μ_k is the mean of the students' scores for the kth subject, ν_i is the ith student's 'verbal intelligence', m_i is the ith student's 'mathematical intelligence', $\ell_{k,j}$ are the factor loadings for the kth subject, for $j = 1, 2$; $\epsilon_{k,i}$ is the difference between the ith student's score in the kth subject and the average score in the kth subject of all students whose levels of verbal and mathematical intelligence are the same as those of the ith student. In matrix notation, we have

$$X = \mu + LF + \epsilon,$$

where X is a $10 \times 1,000$ matrix of observable random variables, μ is a 10×1 column vector of unobservable constants (in this case constants are quantities not differing from one individual student to the next; and random variables are those assigned to individual students; the randomness arises from the random way in which the students are chosen), L is a 10×2 matrix of factor loadings (unobservable constants), F is a $2 \times 1,000$ matrix of unobservable random variables, ϵ is a $10 \times 1,000$ matrix of unobservable random variables.

Observe that by doubling the scale on which 'verbal intelligence', the first component in each column of F, is measured, and simultaneously halving the factor loadings for verbal intelligence makes no difference to the model. Thus, no generality is lost by assuming that the standard deviation of verbal intelligence is 1. Likewise for 'mathematical intelligence'. Moreover, for similar reasons, no generality is lost by assuming the two factors are uncorrelated with each other. The 'errors' ϵ are taken to be independent of each other.

The variances of the 'errors' associated with the 10 different subjects are not assumed to be equal.

Mathematical basis of FA is *principal components analysis* (PCA), which is a technique for simplifying a dataset, by reducing multidimensional datasets to lower dimensions for analysis. Technically speaking, PCA is a *linear transformation*[57] that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA can be used for *dimensionality reduction*[58] in a dataset while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower–order principal components and ignoring higher–order ones. Such low–order components often contain the 'most important' aspects of the data. PCA is also called the (discrete) *Karhunen–Loève transform* (or KLT, named after Kari Karhunen and Michel Loève) or the *Hotelling transform* (in honor of Harold Hotelling[59]). PCA has

[57] Recall that a *linear transformation* (also called *linear map* or *linear operator*) is a function between two vector spaces that preserves the operations of vector addition and scalar multiplication. In the language of abstract algebra, a linear transformation is a *homomorphism of vector spaces*, or a *morphism* in the category of vector spaces over a given field.

Let V and W be vector spaces over the same field K. A function (operator) $f : V \to W$ is said to be a *linear transformation* if for any two vectors $x, y \in V$ and any scalar $a \in K$, the following two conditions are satisfied:

$$\begin{aligned} \text{additivity} &: f(x+y) = f(x) + f(y), \qquad \text{and} \\ \text{homogeneity} &: f(ax) = af(x). \end{aligned}$$

This is equivalent to requiring that for any vectors $x_1, \ldots, x_m$ and scalars $a_1, \ldots, a_m$, the following equality holds:

$$f(a_1x_1 + \cdots + a_mx_m) = a_1f(x_1) + \cdots + a_mf(x_m).$$

[58] Dimensionality reduction in statistics can be divided into two categories: *feature selection* and *feature extraction*.

Feature selection approaches try to find a subset of the original features. Two strategies are *filter* (e.g., information gain) and *wrapper* (e.g., genetic algorithm) approaches. It is sometimes the case that data analysis such as regression or classification can be carried out in the reduced space more accurately than in the original space. On the other hand, feature extraction is applying a mapping of the multidimensional space into a space of fewer dimensions. This means that the original feature space is transformed by applying e.g., a linear transformation via a *principal components analysis*.

Dimensionality reduction is also a phenomenon discussed widely in physics, whereby a physical system exists in three dimensions, but its properties behave like those of a lower–dimensional system.

[59] Harold Hotelling (Fulda, Minnesota, September 29, 1895 - December 26, 1973) was a mathematical statistician. His name is known to all statisticians because of

the distinction of being the optimal linear transformation for keeping the subspace that has largest variance. This advantage, however, comes at the price of greater computational requirement if compared, for example, to the discrete cosine transform. Unlike other linear transforms, the PCA does not have a fixed set of basis vectors. Its basis vectors depend on the data set.

Assuming zero *empirical mean* (the empirical mean of the distribution has been subtracted from the data set), the principal component $\mathbf{w}_1$ of a dataset $\mathbf{x}$ can be defined as

$$\mathbf{w}_1 = \arg\max_{\|\mathbf{w}\|=1} E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^2\right\}.$$

With the first $k-1$ components, the kth component can be found by subtracting the first $k-1$ principal components from $\mathbf{x}$,

$$\hat{\mathbf{x}}_{k-1} = \mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w}_i \mathbf{w}_i^T \mathbf{x},$$

and by substituting this as the new dataset to find a principal component in

$$\mathbf{w}_k = \arg\max_{\|\mathbf{w}\|=1} E\left\{\left(\mathbf{w}^T\hat{\mathbf{x}}_{k-1}\right)^2\right\}.$$

Therefore, the Karhunen–Loève transform is equivalent to finding the *singular value decomposition*[60] of the data matrix $\mathbf{X}$,

---

Hotelling's T–square distribution and its use in statistical hypothesis testing and confidence regions. He also introduced canonical correlation analysis, and is the eponym of *Hotelling's law*, *Hotelling's lemma*, and *Hotelling's rule* in economics.

[60] Recall that in linear algebra, the *singular value decomposition* (SVD) is an important factorization of a rectangular real or complex matrix, with several applications in signal processing and statistics. The SVD can be seen as a generalization of the *spectral theorem*, which says that normal matrices can be unitarily diagonalized using a basis of eigenvectors, to arbitrary, not necessarily square, matrices.

Suppose M is an $m \times n$ matrix whose entries come from the field K, which is either the field of real numbers, or the field of complex numbers. Then there exists a factorization of the form:

$$M = U\Sigma V^*,$$

where U is an $m \times m$ unitary matrix over K, the matrix Σ is $m \times n$ with nonnegative numbers on the diagonal and zeros off the diagonal, and V^* denotes the conjugate transpose of V, an $n \times n$ unitary matrix over K. Such a factorization is called a singular–value decomposition of M.

The matrix V thus contains a set of orthonormal 'input' or 'analyzing' basis vector directions for M. The matrix U contains a set of orthonormal 'output' basis vector directions for M. The matrix Σ contains the singular values, which can be thought of as scalar 'gain controls' by which each corresponding input is multiplied to give a corresponding output. A common convention is to order the values Σ_{ii} in non–increasing fashion. In this case, the diagonal matrix Σ is uniquely determined by M (although the matrices U and V are not).

$$\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T,$$

and then obtaining the reduced–space data matrix $\mathbf{Y}$ by projecting $\mathbf{X}$ down into the reduced space defined by only the first L singular vectors $\mathbf{W_L}$,

$$\mathbf{Y} = \mathbf{W_L}^T\mathbf{X} = \mathbf{\Sigma_L}\mathbf{V_L}^T.$$

The matrix $\mathbf{W}$ of singular vectors of $\mathbf{X}$ is equivalently the matrix $\mathbf{W}$ of eigenvectors of the matrix of observed covariances $\mathbf{C} = \mathbf{X}\mathbf{X}^T$,

$$\mathbf{X}\mathbf{X}^T = \mathbf{W}\mathbf{\Sigma}^2\mathbf{W}^T.$$

The eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset.

Now, FA is performed as PCA[61] with subsequent orthogonal (non–correlated) or oblique (correlated) *factor rotation* for the simplest possible interpretation (see, e.g., [KM78a]).

[61] The alternative FA approach is the so–called *principal factor analysis* (PFA, also called *principal axis factoring*, PAF, and *common factor analysis*, PFA). PFA is a form of factor analysis which seeks the least number of factors which can account for the common variance (correlation) of a set of variables, whereas the more common principal components analysis (PCA) in its full form seeks the set of factors which can account for all the common and unique (specific plus error) variance in a set of variables. PFA uses a PCA strategy but applies it to a correlation matrix in which the diagonal elements are not 1's, as in PCA, but iteratively–derived estimates of the *communalities*.

In addition to PCA and PFA, there are other less–used extraction methods:

1. Image factoring: based on the correlation matrix of predicted variables rather than actual variables, where each variable is predicted from the others using multiple regression.
2. Maximum likelihood factoring: based on a linear combination of variables to form factors, where the parameter estimates are those most likely to have resulted in the observed correlation matrix, using MLE methods and assuming multivariate normality. Correlations are weighted by each variable's uniqueness. (As discussed below, uniqueness is the variability of a variable minus its communality.) MLF generates a *chi–square goodness–of–fit test*. The researcher can increase the number of factors one at a time until a satisfactory goodness of fit is obtained. Warning: for large samples, even very small improvements in explaining variance can be significant by the goodness-of-fit test and thus lead the researcher to select too many factors.
3. Alpha factoring: based on maximizing the reliability of factors, assuming variables are randomly sampled from a universe of variables. All other methods assume cases to be sampled and variables fixed.
4. Unweighted least squares (ULS) factoring: based on minimizing the sum of squared differences between observed and estimated correlation matrices, not counting the diagonal.
5. Generalized least squares (GLS) factoring: based on adjusting ULS by weighting the correlations inversely according to their uniqueness (more unique variables are weighted less). Like MLF, GLS also generates a chi–square goodness–of–fit

FA is used to uncover the latent structure (dimensions) of a set of variables. It reduces attribute space from a larger number of variables to a smaller number of factors and as such is a 'non–dependent' procedure (that is, it does not assume a dependent variable is specified). Factor analysis could be used for any of the following purposes:

1. To reduce a large number of variables to a smaller number of factors for modelling purposes, where the large number of variables precludes modelling all the measures individually. As such, factor analysis is integrated in *structural equation modelling* (SEM),[62] helping create the latent variables modeled by SEM. However, factor analysis can be and is often used on a standalone basis for similar purposes.

test. The researcher can increase the number of factors one at a time until a satisfactory goodness of fit is obtained.

[62] Structural equation modelling (SEM) grows out of and serves purposes similar to multiple regression, but in a more powerful way which takes into account the modelling of interactions, nonlinearities, correlated independents, measurement error, correlated error terms, multiple latent independents each measured by multiple indicators, and one or more latent dependents also each with multiple indicators. SEM may be used as a more powerful alternative to multiple regression, path analysis, factor analysis, time series analysis, and analysis of covariance. That is, these procedures may be seen as special cases of SEM, or, to put it another way, SEM is an extension of the general linear model (GLM) of which multiple regression is a part.

SEM is usually viewed as a confirmatory rather than exploratory procedure, using one of three approaches:

a) Strictly confirmatory approach: A model is tested using SEM goodness–of–fit tests to determine if the pattern of variances and covariances in the data is consistent with a structural (path) model specified by the researcher. However as other unexamined models may fit the data as well or better, an accepted model is only a not–disconfirmed model.
b) Alternative models approach: One may test two or more causal models to determine which has the best fit. There are many goodness–of–fit measures, reflecting different considerations, and usually three or four are reported by the researcher. Although desirable in principle, this AM approach runs into the real-world problem that in most specific research topic areas, the researcher does not find in the literature two well-developed alternative models to test.
c) Model development approach: In practice, much SEM research combines confirmatory and exploratory purposes: a model is tested using SEM procedures, found to be deficient, and an alternative model is then tested based on changes suggested by SEM modification indexes. This is the most common approach found in the literature. The problem with the model development approach is that models confirmed in this manner are post–hoc ones which may not be stable (may not fit new data, having been created based on the uniqueness of an initial dataset). Researchers may attempt to overcome this problem by using a cross–validation strategy under which the model is developed using a calibration data sample and then confirmed using an independent validation sample.

2. To select a subset of variables from a larger set, based on which original variables have the highest correlations with the principal component factors.
3. To create a set of factors to be treated as uncorrelated variables as one approach to handling multi–collinearity in such procedures as multiple regression
4. To validate a scale or index by demonstrating that its constituent items load on the same factor, and to drop proposed scale items which cross–load on more than one factor.
5. To establish that multiple tests measure the same factor, thereby giving justification for administering fewer tests.
6. To identify clusters of cases and/or outliers.
7. To determine network groups by determining which sets of people cluster together.

The so–called *exploratory factor analysis* (EFA) seeks to uncover the underlying structure of a relatively large set of variables. The researcher's à priori assumption is that any indicator may be associated with any factor. This is

---

Regardless of approach, SEM cannot itself draw causal arrows in models or resolve causal ambiguities. Theoretical insight and judgment by the researcher is still of utmost importance.

The SEM process centers around two steps: validating the measurement model and fitting the structural model. The former is accomplished primarily through confirmatory factor analysis, while the latter is accomplished primarily through path analysis with latent variables. One starts by specifying a model on the basis of theory. Each variable in the model is conceptualized as a latent one, measured by multiple indicators. Several indicators are developed for each model, with a view to winding up with at least three per latent variable after confirmatory factor analysis. Based on a large ($n > 100$) representative sample, factor analysis (common factor analysis or principal axis factoring, not principle components analysis) is used to establish that indicators seem to measure the corresponding latent variables, represented by the factors. The researcher proceeds only when the measurement model has been validated. Two or more alternative models (one of which may be the null model) are then compared in terms of *model fit*, which measures the extent to which the covariances predicted by the model correspond to the observed covariances in the data. The so–called modification indices and other coefficients may be used by the researcher to alter one or more models to improve fit.

Advantages of SEM compared to multiple regression include more flexible assumptions (particularly allowing interpretation even in the face of multicollinearity), use of confirmatory factor analysis to reduce measurement error by having multiple indicators per latent variable, the attraction of SEM's graphical modelling interface, the desirability of testing models overall rather than coefficients individually, the ability to test models with multiple dependents, the ability to model mediating variables, the ability to model error terms, the ability to test coefficients across multiple between–subjects groups, and ability to handle difficult data (time series with autocorrelated error, non–normal data, incomplete data).

the most common form of factor analysis. There is no prior theory and one uses factor loadings to intuit the factor structure of the data.

On the other hand, the so–called *confirmatory factor analysis* (CFA) seeks to determine if the number of factors and the loadings of measured (indicator) variables on them conform to what is expected on the basis of pre–established theory. Indicator variables are selected on the basis of prior theory and factor analysis is used to see if they load as predicted on the expected number of factors. The researcher's à priori assumption is that each factor (the number and labels of which may be specified à priori) is associated with a specified subset of indicator variables. A minimum requirement of confirmatory factor analysis is that one hypothesize beforehand the number of factors in the model, but usually also the researcher will posit expectations about which variables will load on which factors (see, e.g., [KM78b]). The researcher seeks to determine, for instance, if measures created to represent a latent variable really belong together.

The *factor loadings*, also called component loadings in PCA, are the correlation coefficients between the variables (rows) and factors (columns) in the *factor matrix*. Analogous to Pearson's r, the squared factor loading is the percent of variance in that variable explained by the factor. To get the percent of variance in all the variables accounted for by each factor, add the sum of the squared factor loadings for that factor (column) and divide by the number of variables (note that the number of variables equals the sum of their variances as the variance of a standardized variable is 1). This is the same as dividing the factor's eigenvalue by the number of variables.

The *factor scores*, also called component scores in PCA, factor scores are the scores of each case (row) on each factor (column). To compute the factor score for a given case for a given factor, one takes the case's standardized score on each variable, multiplies by the corresponding factor loading of the variable for the given factor, and sums these products. Computing factor scores allows one to look for factor outliers. Also, factor scores may be used as variables in subsequent modelling.

Rotation serves to make the output more understandable and is usually necessary to facilitate the interpretation of factors. The sum of eigenvalues is not affected by rotation, but rotation will alter the eigenvalues (and percent of variance explained) of particular factors and will change the factor loadings. Since alternative rotations may explain the same variance (have the same total eigenvalue) but have different factor loadings, and since factor loadings are used to intuit the meaning of factors, this means that different meanings may be ascribed to the factors depending on the rotation – a problem some cite as a drawback to factor analysis. If factor analysis is used, the researcher may wish to experiment with alternative rotation methods to see which leads to the most interpretable factor structure.

Varimax rotation is an orthogonal rotation of the factor axes to maximize the variance of the squared loadings of a factor (column) on all the variables (rows) in a factor matrix, which has the effect of differentiating the original

variables by extracted factor. Each factor will tend to have either large or small loadings of any particular variable. A varimax solution yields results which make it as easy as possible to identify each variable with a single factor. This is the most common rotation option.

The oblique rotations allow the factors to be correlated, and so a factor correlation matrix is generated when oblique is requested. Two most common oblique rotation methods are:

Direct oblimin rotation – the standard method when one wishes a non–orthogonal solution , that is, one in which the factors are allowed to be correlated; this will result in higher eigenvalues but diminished interpretability of the factors; and

Promax rotation – an alternative non–orthogonal rotation method which is computationally faster than the direct oblimin method and therefore is sometimes used for very large datasets.

FA advantages are:

1. Offers a much more objective method of testing intelligence in humans;
2. Allows for a satisfactory comparison between the results of intelligence tests; and
3. Provides support for theories that would be difficult to prove otherwise.

Charles Spearman pioneered the use of factor analysis in the field of psychology and is sometimes credited with the invention of factor analysis. He discovered that schoolchildren's scores on a wide variety of seemingly unrelated subjects were positively correlated, which led him to postulate that a general mental ability, or g, underlies and shapes human cognitive performance. His postulate now enjoys broad support in the field of intelligence research, where it is known as the g theory.

Raymond Cattell expanded on Spearman's idea of a two–factor theory of intelligence after performing his own tests and factor analysis. He used a multi–factor theory to explain intelligence. Cattell's theory addressed alternate factors in intellectual development, including motivation and psychology. Cattell also developed several mathematical methods for adjusting psychometric graphs, such as his 'scree' test and similarity coefficients. His research lead to the development of his theory of fluid and crystallized intelligence. Cattell was a strong advocate of factor analysis and psychometrics. He believed that all theory should be derived from research, which supports the continued use of empirical observation and objective testing to study human intelligence.

Factor Structure and Rotation

Starting with the correlation matrix $\mathbf{R}$ including the number of significant correlations, the goal of exploratory factor analysis (FA) is to detect latent underlying dimensions (i.e., the factor structure) among the set of all manifest variables. Instead of the correlation matrix, the factor analysis can start from the covariance matrix (see Figure 4), which is the symmetrical matrix with

variances of all manifest variables on the main diagonal and their covariances in other matrix cells. For the purpose of the present project the correlation matrix is far more meaningful starting point. Three main applications of factor analytic techniques are (see [CL71, And84, Har75]):

1. to *reduce* the number of manifest variables,
2. to *classify* manifest variables, and
3. to *score* each individual soldier on the latent factor structure.

Factor analysis model expands each of the manifest variables $\mathbf{x}_i$ with the means $\boldsymbol{\mu}_i$ from the data matrix $\mathbf{X} = \{x_{i\alpha}\}$ as a linear vector–function

$$\mathbf{x}_i = \boldsymbol{\mu}_i + \mathbf{L}_{ij}\,\mathbf{f}_j + \mathbf{e}_i, \quad (i = 1, \ldots, n;\ j = 1, \ldots, m) \tag{1.1}$$

where n and m denote the numbers of manifest and latent variables, respectively, $\mathbf{f}_j$ denotes the jth common–factor vector (with zero mean and unity–matrix covariance), $\mathbf{L} = \mathbf{L}_{ij}$ is the matrix of factor loadings l_{ij}, and $\mathbf{e}_i$ corresponds to the ith specific–factor vector (specific variance not explained by the common factors, with zero mean and diagonal–matrix covariance).

That portion of the variance of the ith manifest variable $\mathbf{x}_i$ contributed by the m common factors $\mathbf{f}_j$, the sum of squares of the loadings l_{ij}, is called the ith communality.

Now, in the correlation matrix $\mathbf{R}$ the variances of all variables are equal to 1.0. Therefore, the total variance in that matrix is equal to the number of variables. Extraction of factors is based on the solution of eigenvalue problem, i.e., characteristic equation for the correlation matrix $\mathbf{R}$,

$$\mathbf{R}\mathbf{x}_i = \lambda_i \mathbf{x}_i,$$

where λ_i are eigenvalues of $\mathbf{R}$, representing the variances extracted by the factors, and $\mathbf{x}_i$ now represent the corresponding eigenvectors, representing principal components or factors. The question then is, how many factors do we want to extract? Note that as we extract consecutive factors, they account for less and less variability. The decision of when to stop extracting factors basically depends on when there is only very little 'random' variability left. According to the widely used Kaiser criterion we can retain only factors with eigenvalues greater than 1. In essence this is like saying that, unless a factor extracts at least as much as the equivalent of one original variable, we drop it. The proportion of variance of a particular item that is due to common factors (shared with other items) is called communality. Therefore, an additional task facing us when applying this model is to estimate the communalities for each variable, that is, the proportion of variance that each item has in common with other items. The proportion of variance that is unique to each item is then the respective item's total variance minus the communality. A common starting point is to use the squared multiple correlation of an item with all other items as an estimate of the communality. The correlations between the manifest variables and the principal components are called factor loadings.

The first factor is generally more highly correlated with the variables than the second, third and other factors, as these factors are extracted successively and will account for less and less variance overall.

Therefore, the principal component factor analysis of the sample correlation matrix $\mathbf{R}$ is specified in terms of its $m < n$ eigenvalue–eigenvector pairs $(\lambda_j, \mathbf{x}_j)$ where $\lambda_j \geq \lambda_{j+1}$. The matrix of estimated factor loadings l_{ij} is given by

$$\mathbf{L} = \left[\sqrt{\lambda_1}\mathbf{x}_1 | \sqrt{\lambda_2}\mathbf{x}_2 | \ldots | \sqrt{\lambda_m}\mathbf{x}_m \right].$$

Factor extraction can be performed also by other methods, collectively called *principal factors*, including: (i) Maximum likelihood factors, (ii) Principal axis method, (iii) Centroid method, (iv) Multiple R^2-communalities, and (v) Iterated Minres communalities. However, we shall stick on the principal components because of their obvious eigen–structure.

In any case, matrix of factor loadings $\mathbf{L}$ is determined only up to an orthogonal matrix $\mathbf{O}$. The communalities, given by the diagonal elements of $\mathbf{L}\mathbf{L}^T$ are also unaffected by the choice of $\mathbf{O}$. This ambiguity provides the rationale for 'factor rotation', since orthogonal matrices correspond to 'coordinate' rotations.

We could plot, theoretically, the factor loadings in a $m-$dimensional scatter–plot. In that plot, each variable is represented as a point. In this plot we could rotate the axes in any direction without changing the relative locations of the points to each other; however, the actual coordinates of the points, that is, the factor loadings would of course change. There are various rotational strategies that have been proposed. The goal of all of these strategies is to get a clear pattern of loadings, that is, factors that are somehow clearly marked by high loadings for some variables and low loadings for others. This general pattern is also sometimes referred to as simple structure (a more formalized definition can be found in most standard textbooks). Typical rotational strategies are Varimax, Quartimax, and Equimax (see Anderson, 1984). Basically, the extraction of principal components amounts to a variance maximizing Varimax–rotation of the original space of manifest–variables. We want to get a pattern of loadings on each factor that is as diverse as possible, lending itself to easier interpretation. After we have found the line on which the variance is maximal, there remains some variability around this line. In principal components analysis, after the first factor has been extracted, that is, after the first line has been drawn through the data, we continue and define another line that maximizes the remaining variability, and so on. In this manner, consecutive factors are extracted. Because each consecutive factor is defined to maximize the variability that is not captured by the preceding factor, consecutive factors are independent of each other. Put another way, consecutive factors are uncorrelated or orthogonal to each other.

Basically, the rotation of the matrix of the factor loadings $\mathbf{L}$ represents its post–multiplication, i.e. $\mathbf{L}^* = \mathbf{L}\mathbf{O}$ by the rotation matrix $\mathbf{O}$, which itself resembles one of the matrices included in the classical rotational Lie groups

$SO(m)$ (containing the specific $m-$fold combination of sinuses and cosinuses. The linear factor equation (1.1) represents the orthogonal factor model, provided that vectors $\mathbf{f}_j$ and $\mathbf{e}_i$ are independent (orthogonal to each other, i.e., having zero covariance).

The most frequently used Kaiser's Normal Varimax rotation procedure selects the orthogonal transformation $\mathbf{T}$ that 'spreads out' the squares of the loadings on each factor as much as possible, i.e., maximizes the total 'squared' variance

$$V = \frac{1}{n}\sum_{j=1}^{m}\left[\sum_{i=1}^{n}(l_{ij}^{*})^4 - \frac{1}{n}\left(\sum_{i=1}^{n}(l_{ij}^{*})^2\right)^2\right],$$

where l_{ij}^{*} denote the rotated factor loadings from the rotated factor matrix $\mathbf{L}^{*}$.

Besides orthogonal rotation, there is another concept of oblique (non-orthogonal, or correlated) factors, which could help to achieve more interpretable simple structure. Specifically, computational strategies have been developed to rotate factors so as to best represent clusters of manifest variables, without the constraint of orthogonality of factors. Oblique rotation produces the factor structure made from the smaller set of mutually correlated factors. An oblique rotation to the simple structure corresponds to *nonrigid* rotation of the factor-axes (i.e., principal components) in the factor space such that the rotated axes $\mathbf{l}_{\mathbf{j}}^{*} = \mathbf{L}_{\mathbf{obl}}^{*}$ (no longer perpendicular) pass (nearly) through the clusters of manifest variables. Although the purest mathematical background does not exist for the non–orthogonal factor rotation, the *parsimony principle*: "explain the maximum of the common variability of the data matrix $\mathbf{X} = \{x_{i\alpha}\}$ with the minimum number of factors", is fully developed only in this form of factor analysis, and the factor–correlation matrix $\mathbf{L}_{\mathbf{obl}}^{*}$ resembles the correlation matrix between manifest variables in the latent, factor space with double–reduced number of observables.

The linear factor equation (1.1) becomes now the *oblique factor model*

$$\mathbf{x}_i = \boldsymbol{\mu}_i + \mathbf{L}_{\mathbf{obl}}^{*}\,\mathbf{f}_j + \mathbf{e}_i, \quad (i = 1, \ldots, n;\ j = 1, \ldots, m),$$

where the vectors $\mathbf{f}_j$ and $\mathbf{e}_i$ are interdependent (correlated to each other). With oblique rotation, using common procedures, like Kaiser–Harris Orthoblique, Oblimin, Oblimax, Quartimin, Promax (see [And84]), we could

1. perform a hierarchical (iterated) factor analysis, obtaining second–order factors, third–order factors, etc., finishing with a single general factor (for example using principal component analysis of the factor–correlation matrix $\mathbf{L}_{\mathbf{obl}}^{*}$); and
2. develop the so–called 'cybernetic models': when two factors in the factor–correlation matrix $\mathbf{L}_{\mathbf{obl}}^{*}$ are highly correlated we can assume a linear functional link between them; connecting all correlated factors on the certain hierarchical level, we can make a block–diagram out of them depicting a linear system; this is the real point of the *exploratory* factor analysis.

The factor scores $S_{j\alpha}$ (where j labels factors and α labels individual pilot) are incidental parameters that characterize general performance of the individuals (see [CL71, And84, Har75]). Factor scores with zero mean and unity-matrix covariance are usually automatically evaluated in principal–component, orthogonal and oblique factor analysis, according to the formula:

$$S_{j\alpha} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T(x_{j\alpha} - \bar{x}_{j\alpha}),$$

and replacing $\mathbf{L}$ by $\mathbf{L}^*$, and by $\mathbf{L}^*_{\mathbf{obl}}$, respectively. They represent an objective measure of the general performance of pilot on the battery of psycho–tests.[63]

[63] Here is the Mathematica algorithm for calculating the basic factor structure:
$Mean[x\_] := \frac{Plus@@x}{Length[x]}$;
$Variance[x\_] := \frac{Plus@@(mean[x]-x)^2}{Length[x]}$;
$StDev[x\_] := \sqrt{Variance[x]}$;
$Covar[x1\_, x2\_] := \frac{Plus@@((mean[x1]-x1)((mean[x2]-x2))}{Length[x1]}$;
$Corr[x1\_, x2\_] := \frac{Covar[x1,x2]}{StDev[x1]\,StDev[x2]}$;
$CorrMat[X\_] := Table[Corr[X[[1,j]], X[[1,i]]]//N, \{i,m\}, \{j,m\}]$;
Generate random data–matrix (m variables $\times$ n cases):
$NoVars = 10$; $NoCases = 50$; $m = NoVars$; $n = NoCases$;
$data = Array[x, \{NoCases, NoVars\}]//MatrixForm$;
$Table[x[i,j] = Random[Integer, \{1,5\}], \{i, NoCases\}, \{j, NoVars\}]$;
Print["data = ",data//MatrixForm];
Calculate correlation matrix:
$R = CorrMat[data]$; $Print$["R=",$R//MatrixForm$]
Calculate eigenvalues of the correlation matrix:
$\lambda = Eigenvalues[R]//MatrixForm$
Corresponding eigenvectors:
$vec = Eigenvectors[R]$; $Print[vec//Transpose//MatrixForm]$
Determine significant principal components
according to the criterion $\lambda \geq 2$:
$Print$["PRINCIPAL COMPONENTS"
$\rightarrow \{vec[[1]], vec[[2]]\}//Transpose//MatrixForm$]
Define operator matrix:
$NoFact = 2$; $P = Array[p, NoVars, NoFact]$;
$Table[p[i,j] = 1, \{i, NoVars\}, \{j, NoFact\}]$;
$Table[p[i,j] = 0, \{i, 2, NoVars, 2\}, \{j, 2, NoFact, 2\}]$;
$Table[p[i,j] = 0, \{i, 1, NoVars, 2\}, \{j, 1, NoFact, 2\}]$;
$Print["P = ", P//MatrixForm]$;
Perform oblique rotation:
$Q = Transpose[P]$; $S = R.P$; $G = Q.S$;
$Do[k = \frac{1}{\sqrt{G\|i,i\|}}, \{i, NoFact\}]$;
$F = Sk$; $Z = kG$; $C = Zk$;
$L = Inverse[C]$; $\Phi = F.L$;
Factor structure matrix:
$Print["F = ", F//MatrixForm]$

The factor scores can be used further for multivariate regression in the latent space (instead in the original manifest space) for reducing the number of predictors in the general regression analysis (see [CL71]).

Quantum–Like Correlation and Factor Dynamics

To develop correlation and factor dynamics model, we are using geometrical analogy with *nonrelativistic quantum mechanics* (see [Dir49]). A time dependent state of a quantum system is determined by a normalized (complex), time–dependent, wave psi–function $\psi = \psi(t)$, i.e. a unit Dirac's 'ket' vector $|\psi(t)\rangle$, an element of the Hilbert space $L^2(\psi)$ with a coordinate basis (q^i), under the action of the Hermitian operators, obtained by the procedure of quantization of classical mechanical quantities, for which real eigenvalues are measured . The state–vector $|\psi(t)\rangle$, describing the motion of de Broglie's waves, has a statistical interpretation as the probability amplitude of the quantum system, for the square of its magnitude determines the density of the probability of the system detected at various points of space. The summation over the entire space must yield unity and this is the normalization condition for the psi–function, determining the unit length of the state vector $|\psi(t)\rangle$.

In the coordinate q–representation and the Schrödinger S–picture we consider an action of an evolution operator (in normal units Planck constant $\hbar = 1$)

$$\hat{S} \equiv \hat{S}(t, t_0) = \exp[-i\hat{H}(t - t_0)],$$

i.e., a one–parameter Lie–group of unitary transformations evolving a quantum system. The action represents an exponential map of the system's total energy operator – Hamiltonian $\hat{H} = \hat{H}(t)$. It moves the quantum system from one instant of time, t_0, to some future time t, on the state–vector $|\psi(t)\rangle$, rotating it: $|\psi(t)\rangle = \hat{S}(t, t_0)|\psi(t_0)\rangle$. In this case the Hilbert coordinate basis (q^i) is fixed, so the system operators do not evolve in time, and the system evolution is determined exclusively by the time–dependent Schrödinger equation

$$i\partial_t|\psi(t)\rangle = \hat{H}(t)|\psi(t)\rangle, \qquad (\partial_t = \partial/\partial t), \tag{1.2}$$

with initial condition given at one instant of time t_0 as $|\psi(t_0)\rangle = |\psi\rangle$.

Inter-factor correlation matrix:
$Print[\text{“}C = \text{”}, C//MatrixForm]$
Factor projection matrix:
$Print[\text{“}\Phi = \text{”}, \Phi//MatrixForm]$
Calculate factor scores for individual pilot:
$var[x\_] := x - mean[x];$
$Table[v[i] = var[X[[i]]//N], \{i, n\}];$
$T_F = Transpose[F];\ F_F = Inverse[T_F.F].T_F;$
$Table[F_F.v[i], \{i, n\}]//MatrixForm.$

If the Hamiltonian $\hat{H} = \hat{H}(t)$ does not explicitly depend on time (which is the case with the absence of variables of macroscopic fields), the state vector reduces to the exponential of the system energy:

$$|\psi(t)\rangle \;=\; \exp(-iE(t-t_0)|\psi\rangle,$$

satisfying the time–independent (i.e., stationary) Schrödinger equation

$$\hat{H}|\psi\rangle \;=\; E|\psi\rangle, \tag{1.3}$$

which represents the characteristic equation for the Hamiltonian operator $\hat{H}$ and gives its real eigenvalues (stationary energy states) E_n and corresponding orthonormal eigenfunctions (i.e., probability amplitudes) $|\psi_n\rangle$.

To model the correlation and factor dynamics we start with the characteristic equation for the correlation matrix

$$\mathbf{R}\mathbf{x} = \lambda\mathbf{x},$$

making heuristic analogy with the stationary Schrödinger equation (1.3). This analogy allows a 'physical' interpretation of the correlation matrix $\mathbf{R}$ as an operator of the 'total correlation or covariation energy' of the statistical system (the simulator–test data matrix $\mathbf{X} = \{x_{i\alpha}\}$), eigenvalues λ_n corresponding to the 'stationary energy states', and eigenvectors $\mathbf{x}_n$ resembling 'probability amplitudes' of the system.

So far we have considered one instant of time t_0. Including the time–flow into the stationary Schrödinger equation (1.3) we get the time–dependent Schrödinger equation (1.2) and returning back with our heuristic analogy, we get the basic equation of the n–dimensional correlation dynamics

$$\partial_t \mathbf{x}(t) = \mathbf{R}(t)\,\mathbf{x}_k(t), \tag{1.4}$$

with initial condition at time t_0 given as a stationary manifest–vectors $\mathbf{x}_k(t_0) = \mathbf{x}_k$ $(k = 1, \ldots, n)$.

In more realistic case of 'many' observables (i.e., very big n), instead of the correlation dynamics (1.4), we can use the reduced–dimension factor dynamics, represented by analogous equation in the factor space spanned by the extracted (oblique) factors $\mathbf{F} = \mathbf{f}_i$, with inter–factor–correlation matrix $\mathbf{C} = c_{ij}$ $(i, j = 1, \ldots,$ no. of factors$)$

$$\partial_t \mathbf{f}_i(t) = \mathbf{C}(t)\,\mathbf{f}_i(t), \tag{1.5}$$

subject to initial condition at time t_0 given as stationary vectors $\mathbf{f}_i(t_0) = \mathbf{f}_i$.

Now, according to the fundamental existence and uniqueness theorem for linear autonomous ordinary differential equations, if $A = A(t)$ is an $n \times n$ real matrix, then the initial value problem

$$\partial_t \mathbf{x}(t) = A\mathbf{x}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n,$$

has the unique solution

$$\mathbf{x}(t) = \mathbf{x}_0 \mathrm{e}^{tA}, \qquad \text{for all } t \in \mathbb{R}.$$

Therefore, analytical solutions of our correlation and factor–correlation dynamics equations (1.4) and (1.5) are given respectively by exponential maps

$$\mathbf{x}_k(t) = \mathbf{x}_k \exp[t\,\mathbf{R}],$$
$$\mathbf{f}_i(t) = \mathbf{f}_i \exp[t\,\mathbf{C}].$$

Thus, for each $t \in \mathbb{R}$, the matrix $\mathbf{x} \exp[t\,\mathbf{R}]$, respectively the matrix $\mathbf{f} \exp[t\,\mathbf{C}]$, maps

$$\mathbf{x}_k \mapsto \mathbf{x}_k \exp[t\,\mathbf{R}], \qquad \text{respectively} \qquad \mathbf{f}_i \mapsto \mathbf{f}_i \exp[t\,\mathbf{C}].$$

The sets $g^t_{corr} = \{\exp[t\,\mathbf{R}]\}_{t\in\mathbb{R}}$ and $g^t_{fact} = \{\exp[t\,\mathbf{C}]\}_{t\in\mathbb{R}}$ are 1–parameter families (groups) of linear maps of $\mathbb{R}^n$ into $\mathbb{R}^n$, representing the *correlation flow*, respectively the *factor–correlation flow* of simulator–tests. The linear flows g^t (representing both g^t_{corr} and g^t_{fact}) have two essential properties:

1. identity map: $g^0 = I$, and
2. composition: $g^{t_1+t_2} = g^{t_1} \circ g^{t_2}$.

They partition the state space $\mathbb{R}^n$ into subsets that we call 'correlation orbits', respectively 'factor–correlation orbits', through the initial states $\mathbf{x}_k$, and $\mathbf{f}_i$, of simulator tests, defined respectively by

$$\gamma(\mathbf{x}_k) = \{\mathbf{x}_k g^t | t \in \mathbb{R}\} \qquad \text{and} \qquad \gamma(\mathbf{f}_i) = \{\mathbf{f}_i g^t | t \in \mathbb{R}\}.$$

The correlation orbits can be classified as:

1. If $g^t\mathbf{x}_k = \mathbf{x}_k$ for all $t \in \mathbb{R}$, then $\gamma(\mathbf{x}_k) = \{\mathbf{x}_k\}$ and it is called a *point orbit*. Point orbits correspond to equilibrium points in the manifest and the factor space, respectively.
2. If there exists a $T > 0$ such that $g^T\mathbf{x}_k = \mathbf{x}_k$, then $\gamma(\mathbf{x}_k)$ is called a *periodic orbit*. Periodic orbits describe a system that evolves periodically in time in the manifest and the factor space, respectively.
3. If $g^t\mathbf{x}_k \neq \mathbf{x}_k$ for all $t \neq 0$, then $\gamma(\mathbf{x}_k)$ is called a *non–periodic orbit*.

Analogously, the factor–correlation orbits can be classified as:

1. If $g^t\mathbf{f}_i = \mathbf{f}_i$ for all $t \in \mathbb{R}$, then $\gamma(\mathbf{f}_i) = \{\mathbf{f}_i\}$ and it is called a point orbit. Point orbits correspond to equilibrium points in the manifest and the factor space, respectively.
2. If there exists a $T > 0$ such that $g^T\mathbf{f}_i = \mathbf{f}_i$, then $\gamma(\mathbf{f}_i)$ is called a periodic orbit. Periodic orbits describe a system that evolves periodically in time in the manifest and the factor space, respectively.
3. If $g^t\mathbf{f}_i \neq \mathbf{f}_i$ for all $t \neq 0$, then $\gamma(\mathbf{f}_i)$ is called a non–periodic orbit.

Now, to interpret properly the meaning of (really discrete) time in the correlation matrix $\mathbf{R} = \mathbf{R}(t)$ and factor–correlation matrix $\mathbf{C} = \mathbf{C}(t)$, we can perform a successive time–series $\{t, t + \Delta t, t + 2\Delta t, t + k\Delta t, \cdots\}$ of simulator tests (and subsequent correlation and factor analysis), and discretize our correlation (respectively, factor–correlation) dynamics, to get

$$\begin{aligned} \mathbf{x}_k(t + \Delta t) &= \mathbf{x}_k(0) + \mathbf{R}(t)\, \mathbf{x}_k(t)\Delta t, \qquad \text{and} \\ \mathbf{f}_i(t + \Delta t) &= \mathbf{f}_i(0) + \mathbf{C}(t)\, \mathbf{f}_i(t)\Delta t, \end{aligned}$$

respectively. Finally we can represent the discrete correlation and factor–correlation dynamics in the form of the (computationally applicable) *three–point iterative dynamics equation*, respectively in the manifest space

$$\mathbf{x}_k^{s+1} = \mathbf{x}_k^{s-1} + \mathbf{R}_k^s\, \mathbf{x}_k^s,$$

and in the factor space

$$\mathbf{f}_i^{s+1} = \mathbf{f}_i^{s-1} + \mathbf{C}_i^s\, \mathbf{f}_i^s,$$

in which the time–iteration variable s labels the time occurrence of the simulator tests (and subsequent correlation and factor analysis), starting with the initial state, labelled $s = 0$.

FA–Based Intelligence

In the psychometric view, the concept of intelligence is most closely identified with Spearman's **g**, or Gf ('fluid **g**'). However, psychometricians can measure a wide range of abilities, which are distinct yet correlated. One common view is that these abilities are hierarchically arranged with **g** at the vertex (or top, overlaying all other cognitive abilities).[64]

On the other hand, critics of the psychometric approach, such as Robert Sternberg from Yale, point out that people in the general population have a somewhat different conception of intelligence than most experts. In turn, they argue that the psychometric approach measures only a part of what

[64] Intelligence, IQ, and **g** are distinct terms. As already said above, intelligence is the term used in ordinary discourse to refer to cognitive ability. However, it is generally regarded as too imprecise to be useful for a scientific treatment of the subject. The intelligence quotient (IQ) is an index calculated from the scores on test items judged by experts to encompass the abilities covered by the term intelligence. IQ measures a multidimensional quantity: it is an amalgam of different kinds of abilities, the proportions of which may differ between IQ tests. The dimensionality of IQ scores can be studied by factor analysis, which reveals a single dominant factor underlying the scores on all IQ tests. This factor, which is a hypothetical construct, is called **g**. Variation in **g** corresponds closely to the intuitive notion of intelligence, and thus **g** is sometimes called *general cognitive ability* or *general intelligence*.

is commonly understood as intelligence. Other critics, such as Arthur Eddington,[65] argue that the equipment used in an experiment often determines the results and that proving that e.g., intelligence exists does not prove that current equipment measure it correctly. Sceptics often argue that so much scientific knowledge about the brain is still to be discovered that claiming the conventional IQ test methodology to be infallible is just a small step forward from claiming that *craniometry*[66] was the infallible method for measuring intelligence (which had scientific merits based on knowledge available in the nineteenth century).

A more fundamental criticism is that both the psychometric model used in these studies and the conceptualization of cognitive ability itself are fundamentally off beam. These views were expressed by none other than Charles Spearman, the 'discoverer' of **g** – himself. Thus he wrote: "Every normal man, woman, and child is a genius at something. It remains to discover at what. This must be a most difficult matter, owing to the very fact that it occurs in only a minute proportion of all possible abilities. It certainly cannot be detected by any of the testing procedures at present in current usage. But these procedures are capable, I believe, of vast improvement." In this context he noted that it is more important to ask 'What does this person think about?' than 'How well can he or she think?' Spearman went on to observe that the tests from which his **g** had emerged had no place in schools since they did not reflect the diverse talents of the children and thus deflected teachers from their fundamental educational role, which is to nurture and recognize these diverse talents.

He also noted, as paraphrased here, that the so–called 'cognitive ability' is not primarily cognitive but affective and conative. In constructing meaning out of confusion (Spearman's eductive ability) one first follows feelings that beckon or attract. One then has to engage in 'experimental interactions with the environment' to check out those, largely non–verbal, 'hunches'. This requires determination and persistence — *conation*. Now, all of these are difficult and demanding activities which will only be undertaken whilst one is undertaking activities one cares about. So the first question is: 'What kinds of activity is this person strongly motivated to undertake' (and the kinds of activity which people may be strongly motivated to undertake are legion and mostly unrelated to those assessed in conventional 'intelligence' tests). And the second question is: 'How many of the cumulative and substitutable

[65] Sir Arthur Stanley Eddington, OM (December 28, 1882 — November 22, 1944) was an astrophysicist of the early 20th century. The Eddington limit, the natural limit to the luminosity that can be radiated by accretion onto a compact object, is named in his honor. He is famous for his work regarding the Theory of Relativity. Eddington wrote an article in 1919, Report on the relativity theory of gravitation, which announced Einstein's theory of general relativity to the English–speaking world.

[66] Craniometry is the technique of measuring the bones of the skull. Craniometry was once intensively practiced in anthropology/ethnology.

components of competence required to carry out these activities effectively does this person display whilst carrying out that activity?' So one cannot, in reality, assess a person's intelligence, or even their eductive ability, except in relation to activities they care about. What one sees in e.g., the Raven Progressive Matrices is the cumulative effect of how well they do all these things in relation to a certain sort of task. The problem is that this is not — and cannot be — 'cognitive ability' in any general sense of the word but only in relation to this kind of task. As Roger Sperry[67] has observed, what is neurologically localized is not 'cognitive ability' in any general sense but the emotional predisposition to 'think' about a particular kind of thing (for more details, see e.g., papers of John Raven[68] [Rav02]).

Most experts accept the concept of a single dominant factor of intelligence, general mental ability or **g**, while others argue that intelligence consists of a set of relatively independent abilities [APS98]. The evidence for **g** comes from factor analysis of tests of cognitive abilities. The methods of factor analysis do not guarantee a single dominant factor will be discovered. Other *psychological tests*, which do not measure cognitive ability, such as *personality tests*, generate multiple factors.

Proponents of *multiple–intelligence theories* often claim that **g** is, at best, a measure of academic ability. Other types of intelligence, they claim, might be just as important outside of a school setting. Robert Sternberg has proposed a 'Triarchic Theory of Intelligence'. Howard Gardner's theory of multiple intelligences breaks intelligence down into at least eight different components: logical, linguistic, spatial, musical, kinesthetic, naturalist, intra–personal and

[67] Roger Wolcott Sperry (August 20, 1913 – April 17, 1994) was a neuropsychologist who, together with David Hunter Hubel and Torsten Nils Wiesel, won the 1981 Nobel Prize in Medicine for his work with *split–brain* research. Before Sperry's experiments, some research evidence seemed to indicate that areas of the brain were largely undifferentiated and interchangeable. In his early experiments Sperry challenged this view by showing that after early development circuits of the brain are largely hardwired. In his Nobel–winning work, Sperry separated the *corpus callosum*, the area of the brain used to transfer signals between the right and left hemispheres, to treat epileptics. Sperry and his colleagues then tested these patients with tasks that were known to be dependent on specific hemispheres of the brain and demonstrated that the two halves of the brain may each contain consciousness. In his words, each hemisphere is "... indeed a conscious system in its own right, perceiving, thinking, remembering, reasoning, willing, and emoting, all at a characteristically human level, and . . . both the left and the right hemisphere may be conscious simultaneously in different, even in mutually conflicting, mental experiences that run along in parallel." This research contributed greatly to understanding the lateralization of brain functions. In 1989, Sperry also received National Medal of Science.

[68] John Carlyle Raven first published his Progressive Matrices in the United Kingdom in 1938. His three sons established Scotland–based test publisher JC Raven Ltd. in 1972. In 2004, Harcourt Assessment, Inc. a division of Harcourt Education acquired JC Raven Ltd.

inter–personal intelligences. Daniel Goleman and several other researchers have developed the concept of *emotional intelligence* and claim it is at least as important as more traditional sorts of intelligence. These theories grew from observations of human development and of brain injury victims who demonstrate an acute loss of a particular cognitive function (e.g., the ability to think numerically, or the ability to understand written language), without showing any loss in other cognitive areas.

In response, **g** theorists have pointed out that **g**'s *predictive validity*[69] has been repeatedly demonstrated, for example in predicting important non–academic outcomes such as job performance, while no multiple–intelligences theory has shown comparable validity. Meanwhile, they argue, the relevance, and even the existence, of multiple intelligences have not been borne out when actually tested [Hun01]. Furthermore, **g** theorists contend that proponents of multiple–intelligences (see, e.g., [Ste95]) have not disproved the existence of a general factor of intelligence [Kli00]. The fundamental argument for a general factor is that test scores on a wide range of seemingly unrelated cognitive ability tests (such as sentence completion, arithmetic, and memorization) are positively correlated: people who score highly on one test tend to score highly on all of them, and g thus emerges in a factor analysis. This suggests that the tests are not unrelated, but that they all tap a common factor.

Cognitive vs. Not–Cognitive Intelligence

Clearly, biologically realized 'cognitive intelligence' is the most complex property of human mind and can be perceived only by itself. Our problem is what we call or may call cognitive intelligence. From the formal, computational perspective, cognitive intelligence is one of ill defined concepts. Its definitions are immersed in numerous scientific contexts and mirrors their historical evolutions, as well as, different 'interests' of researchers. Its weakness is usually based on its abstract multifaces image and, on the other hand, a universal utility character.

[69] In psychometrics, *predictive validity* is the extent to which a scale predicts scores on some criterion measure. For example, the validity of a cognitive test for job performance is the correlation between test scores and, say, supervisor performance ratings. Such a cognitive test would have predictive validity if the observed correlation were statistically significant. Predictive validity shares similarities with concurrent validity in that both are generally measured as correlations between a test and some criterion measure. In a study of concurrent validity the test is administered at the same time as the criterion is collected. This is a common method of developing validity evidence for employment tests: A test is administered to incumbent employees, then a rating of those employees' job performance is obtained (often, as noted above, in the form of a supervisor rating). Note the possibility for restriction of range both in test scores and performance scores: The incumbent employees are likely to be a more homogeneous and higher performing group than the applicant pool at large.

The classical behavioral/biologists definition of intelligence reads: "Intelligence is the ability to adapt to new conditions and to successfully cope with life situations." This definition seems to be the best, but 'intelligence' here depends on available physical tools and specific life experience (individual hidden knowledge, preferences and access to information), therefore it is not enough selective to be measured, compared or designed. In general, cognitive intelligence is a human–like intelligence. Unfortunately there are many opinions what human–like intelligence means. For example, (i) cognitive intelligence uses a human mental introspective experience for the modelling of intelligent system thinking; and (ii) cognitive intelligence may use brain models to extract brain's intelligence property.

Therefore, cognitive intelligence can be seen as a product of human self–conscious recognition of efficient mental processes, defined a'priori as intelligent. In order to get a consensus on the notion of cognitive intelligence is useful to have an agreement on which intelligence is not cognitive. A not–cognitive intelligence could be considered as an intelligence being developed using not human analogies; e.g., it is possible to construct very different models of flying objects starting from the observation of storks, balloons, beetles or clouds – maybe this observation can be useful.

The difference between human and artificial intelligence theories is similar to the difference between a birds theory of fly and the airplanes fly theory, the both can lead to a more general theory of fly but this last needs a goal–oriented and a higher abstraction level of the conceptualization/ontology.

According to the *TOGA meta–theory paradigms*,[70] for scientific and practical modelling purposes, it is reasonable to separate conceptually the following five concepts: *information, knowledge, preferences, intelligence and emotions.* If properly defined, all of them can be independently identified and designed.

Such conceptual modularity should enable to construct: *emotional intelligence, social intelligence, skill intelligence, organizational intelligence,* and many other X–intelligences, where X denotes a type of knowledge, preferences or a carrier system involved.

[70] According to the *top–down object–based goal–oriented approach* (TOGA) standard, the Information–Preferences–Knowledge *cognitive architecture* consists of:

Data: everything what is/can be processed/transformed in computational and mental processes. Concept data is included in the ontology of 'elaborators', such as developers of methods, programmers and other computation service people. In this sense, data is a relative term and exists only in the couple (data, processing).

Information: data which represent a specific property of the domain of human or artificial agent's activity (such as: addresses, tel. numbers, encyclopedic data, various lists of names and results of measurements). Every information has always a source domain. It is a relative concept. Information is a concept from the ontology of modeler/problem–solver/decision–maker.

Knowledge: every abstract property of human/artificial agent which has ability to process/transform a quantitative/qualitative information into other information, or into another knowledge. It includes: instructions, emergency procedures,

For example, business intelligence and emotional intelligence, rather are applications of intelligence either for business activities or for the second, under emotional/(not conscious) constrains and 'biological requests'.

In the above context, an *abstract intelligent agent* can be considered as a functional kernel of any natural or artificial intelligent system.

Intelligence and Cognitive Development

Although there is no general *theory of cognitive development*, the most historically influential theory was developed by Jean Piaget.[71] *Piaget theory*

exploitation/user manuals, scientific materials, models and theories. Every knowledge has its reference domain where it is applicable. It has to include the source domain of the processed information. It is a relative concept.

Preference: an ordered relation among two properties of the domain of activity of a *cognitive agent*, it indicates a property with higher utility. Preference relations serve to establish an intervention goal of an agent. Cognitive preferences are relative. A preference agent which manages preferences of an intelligent agent can be external or its internal part.

Goal: a hypothetical state of the domain of activity which has maximal utility in a current situation. Goal serves to the choice and activate proper knowledge which process new information.

Document: a passive carrier of knowledge, information and/or preferences (with different structures), comprehensive for humans, and it has to be recognized as valid and useful by one or more human organizations, it can be physical or electronic.

Computer Program: (i) from the modelers and decision-makers perspective: an active carrier of different structures of knowledge expressed in computer languages and usually focused on the realization of predefined objectives (a design-goal). It may include build-in preferences and information and/or request specific IPK as data. (ii) from the software engineers perspective: a data-processing tool (more precise technical def. you may find on the Web).

[71] Jean Piaget (August 9, 1896 – September 16, 1980) was a Swiss natural scientist and developmental psychologist, well known for his work studying children and his theory of cognitive development. Piaget served as professor of psychology at the University of Geneva from 1929 to 1975 and is best known for reorganizing cognitive development theory into a series of stages, expanding on earlier work from James Baldwin: four levels of development corresponding roughly to (1) infancy, (2) pre–school, (3) childhood, and (4) adolescence. Each stage is characterized by a general cognitive structure that affects all of the child's thinking (a structuralist view influenced by philosopher Immanuel Kant). Each stage represents the child's understanding of reality during that period, and each but the last is an inadequate approximation of reality. Development from one stage to the next is thus caused by the accumulation of errors in the child's understanding of the environment; this accumulation eventually causes such a degree of cognitive disequilibrium that thought structures require reorganising. For his development of the theory, Piaget was awarded the Erasmus Prize.

provided many central concepts in the field of developmental psychology. His theory concerned the growth of intelligence, which for Piaget meant the ability to more accurately represent the world, and perform logical operations on representations of concepts grounded in the world. His theory concerns the emergence and acquisition of schemata, schemes of how one perceives the world, in 'developmental stages', times when children are acquiring new ways of mentally representing information. Piaget theory is considered 'constructivist, meaning that, unlike nativist theories (which describe cognitive development as the unfolding of innate knowledge and abilities) or empiricist theories (which describe cognitive development as the gradual acquisition of knowledge through experience), asserts that we construct our cognitive abilities through self–motivated action in the world.

The four development stages are described in Piaget's theory as:

1. Sensorimotor stage: from birth to age 2 years (children experience the world through movement and senses)
2. Preoperational stage: from ages 2 to 7(acquisition of motor skills)
3. Concrete operational stage: from ages 7 to 11 (children begin to think logically about concrete events)
4. Formal Operational stage: after age 11 (development of abstract reasoning).

These chronological periods are approximate, and in light of the fact that studies have demonstrated great variation between children, cannot be seen as rigid norms. Furthermore, these stages occur at different ages, depending upon the domain of knowledge under consideration. The ages normally given for the stages, then, reflect when each stage tends to predominate, even though one might elicit examples of two, three, or even all four stages of thinking at the same time from one individual, depending upon the domain of knowledge and the means used to elicit it. Despite this, though, the principle holds that within a domain of knowledge, the stages usually occur in the same chronological order. Thus, there is a somewhat subtler reality behind the normal characterization of the stages as described above. The reason for the invariability of sequence derives from the idea that knowledge is not simply acquired from outside the individual, but it is constructed from within. This idea has been extremely influential in pedagogy, and is usually termed constructivism. Once knowledge is constructed internally, it is then tested against reality the same way a scientist tests the validity of hypotheses. Like a scientist, the individual learner may discard, modify, or reconstruct knowledge based on its utility in the real world. Much of this construction (and later reconstruction) is in fact done subconsciously. Therefore, Piaget's four stages actually reflect four types of thought structures. The chronological sequence is inevitable, then, because one structure may be necessary in order to construct the next level, which is simpler, more generalizable, and more powerful. It's a little like saying that you need to form metal into parts in order to build machines, and then coordinate machines in order to build a factory.

Piaget divided schemes that children use to understand the world through four main stages, roughly correlated with and becoming increasingly sophisticated with age:

1. Sensorimotor stage (years 0–2),
2. Preoperational stage (years 2–7),
3. Concrete operational stage (years 7–11), and
4. Formal operational stage (years 11–adulthood).

Sensorimotor Stage

Infants are born with a set of congenital reflexes, according to Piaget, as well as a drive to explore their world. Their initial schemas are formed through differentiation of the congenital reflexes (see assimilation and accommodation, below).

The sensorimotor stage is the first of the four stages. According to Piaget, this stage marks the development of essential spatial abilities and understanding of the world in six sub–stages:

1. The first sub–stage occurs from birth to six weeks and is associated primarily with the development of reflexes. Three primary reflexes are described by Piaget: sucking of objects in the mouth, following moving or interesting objects with the eyes, and closing of the hand when an object makes contact with the palm (palmar grasp). Over these first six weeks of life, these reflexes begin to become voluntary actions; for example, the palmar reflex becomes intentional grasping.
2. The second sub–stage occurs from six weeks to four months and is associated primarily with the development of habits. Primary circular reactions or repeating of an action involving only ones own body begin. An example of this type of reaction would involve something like an infant repeating the motion of passing their hand before their face. Also at this phase, passive reactions, caused by classical or operant conditioning, can begin.
3. The third sub–stage occurs from four to nine months and is associated primarily with the development of coordination between vision and prehension. Three new abilities occur at this stage: intentional grasping for a desired object, secondary circular reactions, and differentiations between ends and means. At this stage, infants will intentionally grasp the air in the direction of a desired object, often to the amusement of friends and family. Secondary circular reactions, or the repetition of an action involving an external object begin; for example, moving a switch to turn on a light repeatedly. The differentiation between means also occurs. This is perhaps one of the most important stages of a child's growth as it signifies the dawn of logic. Towards the late part of this sub–stage infants begin to have a sense of object permanence, passing the A–not–B error test.
4. The fourth sub-stage occurs from nine to twelve months and is associated primarily with the development of logic and the coordination between

means and ends. This is an extremely important stage of development, holding what Piaget calls the 'first proper intelligence'. Also, this stage marks the beginning of goal orientation, the deliberate planning of steps to meet an objective.
5. The fifth sub–stage occurs from twelve to eighteen months and is associated primarily with the discovery of new means to meet goals. Piaget describes the child at this juncture as the 'young scientist', conducting pseudo–experiments to discover new methods of meeting challenges.
6. The sixth sub–stage is associated primarily with the beginnings of insight, or true creativity. This marks the passage into the preoperational stage.

Preoperational Stage

The Preoperational stage is the second of four stages of cognitive development. By observing sequences of play, Piaget was able to demonstrate that towards the end of the second year a qualitatively quite new kind of psychological functioning occurs. Operation in Piagetian theory is any procedure for mentally acting on objects. The hallmark of the preoperational stage is sparse and logically inadequate mental operations.

According to Piaget, the Sensorimotor stage of development is followed by this stage (2–7 years), which includes the following five processes:

1. Symbolic functioning, which is characterised by the use of mental symbols words or pictures which the child uses to represent something which is not physically present.
2. Centration, which is characterized by a child focusing or attending to only one aspect of a stimulus or situation. For example, in pouring a quantity of liquid from a narrow beaker into a shallow dish, a preschool child might judge the quantity of liquid to have decreased, because it is 'lower', that is, the child attends to the height of the water, but not to the compensating increase in the diameter of the container.
3. Intuitive thought, which occurs when the child is able to believe in something without knowing why she or he believes it.
4. Egocentrism, which is a version of centration, this denotes a tendency of child to only think from own point of view.
5. Inability to Conserve; Through Piaget's conservation experiments (conservation of mass, volume and number) Piaget concluded that children in the preoperational stage lack perception of conservation of mass, volume, and number after the original form has changed. For example, a child in this phase will believe that a string of beads set up in a 'O–O–O–O–O' pattern will have the same number of beads as a string which has a 'O–O–O–O–O' pattern, because they are the same length, or that a tall, thin 8-ounce cup has more liquid in it than a wide, fat 8–ounce cup.

Concrete Operational Stage

The concrete operational stage is the third of four stages of cognitive development in Piaget's theory. This stage, which follows the Preoperational stage and occurs from the ages of 7 to 11, is characterized by the appropriate use of logic. The six important processes during this stage are:

1. Decentering, where the child takes into account multiple aspects of a problem to solve it. For example, the child will no longer perceive an exceptionally wide but short cup to contain less than a normally-wide, taller cup.
2. Reversibility, where the child understands that numbers or objects can be changed, then returned to their original state. For this reason, a child will be able to rapidly determine that $4 + 4$ which they can answer to be 8, minus 4 will equal four, the original quantity.
3. Conservation: understanding that quantity, length or number of items is unrelated to the arrangement or appearance of the object or items. For instance, when a child is presented with two equally–sized, full cups they will be able to discern that if water is transferred to a pitcher it will conserve the quantity and be equal to the other filled cup.
4. Serialisation: the ability to arrange objects in an order according to size, shape, or any other characteristic. For example, if given different–shaded objects they may make a color gradient.
5. Classification: the ability to name and identify sets of objects according to appearance, size or other characteristic, including the idea that one set of objects can include another. A child is no longer subject to the illogical limitations of animism (the belief that all objects are animals and therefore have feelings).
6. Elimination of Egocentrism: the ability to view things from another's perspective (even if they think incorrectly). For instance, show a child a comic in which Jane puts a doll under a box, leaves the room, and then Jill moves the doll to a drawer, and Jane comes back; a child in this stage will not say that Jane will think the doll is in the drawer.

Formal Operational Stage

The formal operational stage is the fourth and final of the stages of cognitive development of Piaget's theory. This stage, which follows the Concrete Operational stage, commences at around 11 years of age (puberty) and continues into adulthood. It is characterized by acquisition of the ability to think abstractly and draw conclusions from the information available. During this stage the young adult functions in a cognitively normal manner and therefore is able to understand such things as love, 'shades of gray', and values. Lucidly, biological factors may be traced to this stage as it occurs during puberty and marks the entering into adulthood in physiologically, cognitive, moral (Kohlberg), psychosexual (Freud), and social development (Erikson).

Many people do not successfully complete this stage, but mostly remain in concrete operations.

Psychophysics

Recall that *psychophysics* is a subdiscipline of psychology, founded in 1860 by Gustav Fechner[72] with the publication of 'Elemente der Psychophysik', dealing with the relationship between physical stimuli and their subjective correlates, or percepts. Fechner described research relating physical stimuli with how they are perceived and set out the philosophical foundations of the field. Fechner wanted to develop a theory that could relate matter to the mind, by describing the relationship between the world and the way it is perceived (Snodgrass, 1975). Fechner's work formed the basis of psychology as a science. Wilhelm Wundt, the founder of the first laboratory for psychological research, built upon Fechner's work.

The *Weber–Fechner law* attempts to describe the relationship between the physical magnitudes of stimuli and the perceived intensity of the stimuli.

[72] Gustav Theodor Fechner (April 19, 1801 – November 28, 1887), was a German experimentle psychologist. A pioneer in experimental psychology.

Fechner's epoch–faking work was his Elemente der Psychophysik (1860). He starts from the Spinozistic thought that bodily facts and conscious facts, though not reducible one to the other, are different sides of one reality. His originality lies in trying to discover an exact mathematical relation between them. The most famous outcome of his inquiries is the law known as *Weber–Fechner law* which may be expressed as follows: "In order that the intensity of a sensation may increase in arithmetical progression, the stimulus must increase in geometrical progression." Though holding good within certain limits only, the law has been found immensely useful. Unfortunately, from the tenable theory that the intensity of a sensation increases by definite additions of stimulus, Fechner was led on to postulate a unit of sensation, so that any sensations might be regarded as composed of n units. Sensations, he argued, thus being representable by numbers, psychology may become an 'exact' science, susceptible of mathematical treatment.

His general formula for getting at the number of units in any sensation is $S = c \log R$, where S stands for the sensation, R for the stimulus numerically estimated, and c for a constant that must be separately determined by experiment in each particular order of sensibility. This reasoning of Fechner's has given rise to a great mass of controversy, but the fundamental mistake in it is simple. Though stimuli are composite, sensations are not. "Every sensation," says William James, "presents itself as an indivisible unit; and it is quite impossible to read any clear meaning into the notion that they are masses of units combined." Still, the idea of the exact measurement of sensation has been a fruitful one, and mainly through his influence on Wilhelm Wundt, Fechner was the father of that 'new' psychology of laboratories which investigates human faculties with the aid of exact scientific apparatus.

Ernst Weber[73] was one of the first people to approach the study of the human response to a physical stimulus in a quantitative fashion. Gustav Fechner later offered an elaborate theoretical interpretation of Weber's findings, which he called simply Weber's law, though his admirers made the law's name a hyphenate. Fechner believed that Weber had discovered the fundamental principle of mind/body interaction, a mathematical analog of the function Rene Descartes once assigned to the pineal gland.

In one of his classic experiments, Weber gradually increased the weight that a blindfolded man was holding and asked him to respond when he first felt the increase. Weber found that the response was proportional to a relative increase in the weight. That is to say, if the weight is 1 kg, an increase of a few grams will not be noticed. Rather, when the mass is increased by a certain factor, an increase in weight is perceived. If the mass is doubled, the threshold is also doubled. This kind of relationship can be described by a linear ordinary differential equation as,

$$dp = k\frac{dS}{S},$$

where dp is the differential change in perception, dS is the differential increase in the stimulus and S is the stimulus at the instant. A constant factor k is to be determined experimentally. Integrating the above equation gives: $p = k \ln S + C$, where C is the constant of integration. To determine C, we can put $p = 0$, which means no perception; then we get, $C = -k \ln S_0$,where S_0 is that threshold of stimulus below which it is not perceived at all. In this way, we get the solution

$$p = k \ln \frac{S}{S_0}.$$

Therefore, the relationship between stimulus and perception is logarithmic. This logarithmic relationship means that if a stimulus varies as a geometric progression (i.e. multiplied by a fixed factor), the corresponding perception is altered in an arithmetic progression (i.e. in additive constant amounts). For example, if a stimulus is tripled in strength (i.e, 3×1), the corresponding perception may be two times as strong as its original value (i.e., $1 + 1$). If the stimulus is again tripled in strength (i.e., $3 \times 3 \times 1$), the corresponding perception will be three times as strong as its original value (i.e., $1 + 1 + 1$). Hence, for multiplications in stimulus strength, the strength of perception

[73] Ernst Heinrich Weber (Wittenberg, June 24, 1795 – January 26, 1878) was a German physician who is considered a founder of experimental psychology. Weber studied medicine at Wittenberg University. In 1818 he was appointed Associate Professor of comparative anatomy at Leipzig University, where he was made a Fellow Professor of anatomy and physiology in 1821.

Around 1860 Weber worked with Gustav Fechner on psychophysics, during which time he formulated Weber's Law. In 1866 Weber retired as professor of physiology and also as professor of anatomy in 1871. Around this time he and his brother, Eduard Weber, discovered the inhibitory power of the vagus nerve.

only adds. This logarithmic relationship is valid, not just for the sensation of weight, but for other stimuli and our sensory perceptions as well.

In case of vision, we have that the eye senses brightness logarithmically. Hence stellar magnitude is measured on a logarithmic scale. This magnitude scale was invented by the ancient Greek astronomer Hipparchus in about 150 B.C. He ranked the stars he could see in terms of their brightness, with 1 representing the brightest down to 6 representing the faintest, though now the scale has been extended beyond these limits. An increase in 5 magnitudes corresponds to a decrease in brightness by a factor 100.

In case of sound, we have still another logarithmic scale is the decibel scale of sound intensity. And yet another is pitch, which, however, differs from the other cases in that the physical quantity involved is not a 'strength'. In the case of perception of pitch, humans hear pitch in a logarithmic or geometric ratio–based fashion: For notes spaced equally apart to the human ear, the frequencies are related by a multiplicative factor. For instance, the frequency of corresponding notes of adjacent octaves differ by a factor of 2. Similarly, the perceived difference in pitch between 100 Hz and 150 Hz is the same as between 1000 Hz and 1500 Hz. Musical scales are always based on geometric relationships for this reason. Notation and theory about music often refers to pitch intervals in an additive way, which makes sense if one considers the logarithms of the frequencies, as $\log(a \times b) = \log a + \log b$.

Psychophysicists usually employ experimental stimuli that can be objectively measured, such as pure tones varying in intensity, or lights varying in luminance. All the senses have been studied:vision, hearing, touch (including skin and enteric perception), taste, smell, and the sense of time. Regardless of the sensory domain, there are three main topics in the psychophysical classification scheme: absolute thresholds, discrimination thresholds, and scaling.

The most common use of psychophysics is in producing scales of human experience of various aspects of physical stimuli. Take for an example the physical stimulus of frequency of sound. Frequency of a sound is measured in Hertz (Hz), cycles per second. But human experience of the frequencies of sound is not the same as the frequencies. For one thing, there is a frequency below which no sounds can be heard, no matter how intense they are (around 20 Hz depending on the individual) and there is a frequency above which no sounds can be heard, no matter how intense they are (around 20,000 Hz, again depending on the individual). For another, doubling the frequency of a sound (e.g., from 100 Hz to 200 Hz) does not lead to a doubling of experience. The perceptual experience of the frequency of sound is called pitch, and it is measured by psychophysicists in mels.

More analytical approaches allow the use of psychophysical methods to study neurophysiological properties and sensory processing mechanisms. This is of particular importance in human research, where other (more invasive) methods are not used due to ethical reasons. Areas of investigation include sensory thresholds, methods of measurement of sensitivity, and signal detection theory.

Perception is the process of acquiring, interpreting, selecting, and organizing sensory information. Methods of studying perception range from essentially biological or physiological approaches, through psychological approaches to the often abstract 'thought–experiments' of mental philosophy.

Experiments in psychophysics seek to determine whether the subject can detect a stimulus, identify it, differentiate between it and another stimulus, and describe the magnitude or nature of this difference [Sno75]. Often, the classic methods of experimentation are argued to be inefficient. This is because a lot of sampling and data has to be collected at points of the psychometric function that is known (the tails). Staircase procedures can be used to quickly estimate threshold. However, the cost of this efficiency, is that we do not get the same amount of information regarding the *psychometric function* as we can through classical methods; e.g., we cannot extract an estimate of the slope (derivative) of the function.

A psychometric function describes the relationship between a parameter of a physical stimulus and the responses of a person who has to decide about a certain aspect of that stimulus. The psychometric function usually resembles a sigmoid function with the percentage of correct responses (or a similar value) displayed on the ordinate and the physical parameter on the abscissa. If the stimulus parameter is very far towards one end of its possible range, the person will always be able to respond correctly. Towards the other end of the range, the person never perceives the stimulus properly and therefore the probability of correct responses is at chance level. In between, there is a transition range where the subject has an above–chance rate of correct responses, but does not always respond correctly. The inflection point of the sigmoid function or the point at which the function reaches the middle between the chance level and 100% is usually taken as sensory threshold. A common example is visual acuity testing with an eye chart. The person sees symbols of different sizes (the size is the relevant physical stimulus parameter) and has to decide which symbol it is. Usually, there is one line on the chart where a subject can identify some, but not all, symbols. This is equal to the transition range of the psychometric function and the sensory threshold corresponds to visual acuity.

On the other hand, a *sensory threshold* is a theoretical concept which states: "A stimulus that is less intense than the sensory threshold will not elicit any sensation." Whilst the concept can be applied to all senses, it is most commonly applied to the detection and perception of flavours and aromas. Several different sensory thresholds have been defined:

1. Absolute threshold: the lowest level at which a stimulus can be detected.
2. Recognition threshold: the level at which a stimulus can not only be detected but also recognised.
3. Differential threshold: the level at which an increase in a detected stimulus can be perceived.
4. Terminal threshold: the level beyond which a stimulus is no longer detected.

In other words, a threshold is the point of intensity at which the participant can just detect the presence of, or difference in, a stimulus. Stimuli with intensities below the threshold are considered not detectable, however stimuli at values close to threshold will often be detectable some proportion of the time. Due to this, a threshold is considered to be the point at which a stimulus, or change in a stimulus, is detected some proportion p of the time. An absolute threshold is the level of intensity of a stimulus at which the subject is able to detect the presence of the stimulus some proportion of the time (a p level of 50% is often used). An example of an absolute threshold is the number of hairs on the back of one's hand that must be touched before it can be felt, a participant may be unable to feel a single hair being touched, but may be able to feel two or three as this exceeds the threshold. A difference threshold is the magnitude of the difference between two stimuli of differing intensities that the participant is able to detect some proportion of the time (again, 50% is often used). To test this threshold, several difference methods are used. The subject may be asked to adjust one stimulus until it is perceived as the same as the other, may be asked to describe the magnitude of the difference between two stimuli, or may be asked to detect a stimulus against a background. Absolute and difference thresholds are sometimes considered similar because there is always background noise interfering with our ability to detect stimuli, however study of difference thresholds still occurs, for example in pitch discrimination tasks (see [Sno75]).

The *sensory analysis* applies principles of experimental design and statistical analysis to the use of human senses (sight, smell, taste, touch and hearing) for the purposes of evaluating consumer products. The discipline requires panels of human assessors, on whom the products are tested, and recording the responses made by them. By applying statistical techniques to the results it is possible to make inferences and insights about the products under test. Most large consumer goods companies have departments dedicated to sensory analysis. Sensory Analysis can generally be broken down into three sub–sections:

1. Effective Testing (dealing with objective facts about products);
2. Affective Testing (dealing with subjective facts such as preferences); and
3. Perception (the biochemical and psychological aspects of sensation).

The *signal detection theory* (SDT) is a means to quantify the ability to discern between signal and noise. It has applications in many fields such as quality control, telecommunications, and psychology (see [Abd06]). The concept is similar to the signal to noise ratio used in the sciences, and it is also usable in alarm management, where it is important to separate important events from background noise. According to the theory, there are a number of psychological determiners of how we will detect a signal, and where our threshold levels will be. Experience, expectations, physiological state (e.g, fatigue) and other factors affect thresholds. For instance, a sentry in wartime will likely detect fainter stimuli than the same sentry in peacetime. SDT is used when psychologists want to measure the way we make decisions under

conditions of uncertainty, such as how we would perceive distances in foggy conditions. SDT assumes that 'the decision maker is not a passive receiver of information, but an active decision–maker who makes difficult perceptual judgements under conditions of uncertainty'. In foggy circumstances, we are forced to decide how far an object is away from us based solely upon visual stimulus which is impaired by the fog. Since the brightness of the object, such as a traffic light, is used by the brain to discriminate the distance of an object, and the fog reduces the brightness of objects, we perceive the object to be much further away than it actually is. To apply signal detection theory to a data set where stimuli were either present or absent, and the observer categorized each trial as having the stimulus present or absent, the trials are sorted into one of four categories, depending upon the stimulus and response:

	Respond 'Absent'	Respond 'Present'
Stimulus Present	Miss	Hit
Stimulus Absent	Correct Rejection	False Alarm

1.1.2 Human Problem Solving

Beginning in the 1970s, researchers became increasingly convinced that empirical findings and theoretical concepts derived from simple laboratory tasks did not necessarily generalize to more complex, real–life problems. Even worse, it appeared that the processes underlying creative problem solving in different domains differed from each other [Ste95]. These realizations have led to rather different responses in North America and in Europe.

In North America, initiated by the work of Herbert Simon on learning by doing in semantically rich domains (see, e.g., [AS79, BS77]), researchers began to investigate problem solving separately in different natural knowledge domains – such as physics, writing, or chess playing – thus relinquishing their attempts to extract a global theory of problem solving (see, e.g., [SF91]). Instead, these researchers have frequently focused on the development of problem solving within a certain domain, that is on the development of expertise (see, e.g., [ABR85], [CS73]; [CFG81]).

Areas that have attracted rather intensive attention in North America include such diverse fields as: reading [SC91], writing [BBS91], calculation [SM91], political decision making [VWL91], managerial problem solving [Wag91], lawyers' reasoning [ALL91], personal problem solving [HK87], mathematical problem solving [Pol45, Sch85], mechanical problem solving [Heg91], problem solving in electronics [LL91], computer skills [Kay91], game playing [FS91], and social problem solving [D'Zur86].

In particular, George Pólya's 1945 book 'How to Solve It' [Pol45], is a small volume describing methods of problem–solving. It suggests the following steps when solving a mathematical problem:

1. First, you have to understand the problem.
2. After understanding, then make a plan.

Heuristic	**Informal Description**	**Formal analogue**
Analogy	can you find a problem analogous to your problem and solve that?	Map
Generalization	can you find a problem more general than your problem ...?	Generalization
Induction	can you solve your problem by deriving a generalization from some examples?	Induction
Variation of the Problem	can you vary or change your problem to create a new problem (or set of problems) whose solution(s) will help you solve your original problem?	Search
Auxiliary Problem	can you find a subproblem or side problem whose solution will help you solve your problem?	Subgoal
Here is a problem related to yours and solved before	can you find a problem related to yours that has already been solved and use that to solve your problem?	Pattern recognition Pattern matching
Specialization	can you find a problem more specialized?	Specialization
Decomposing and Recombining	can you decompose the problem and "recombine its elements in some new manner"?	Divide and conquer
Working backward	can you start with the goal and work backwards to something you already know?	Backward chaining
Draw a Figure	can you draw a picture of the problem?	Diagrammatic Reasoning
Auxiliary Elements	can you add some new element to your problem to get closer to a solution?	Extension

3. Carry out the plan.
4. Look back on your work. How could it be better?

If this technique fails, Polya advises: "If you cannot solve a problem, then there is an easier problem you can solve: find it." Or, "If you cannot solve the proposed problem try to solve first some related problem. Could you imagine a more accessible related problem?"

His small book contains a dictionary–style set of heuristics, many of which have to do with generating a more accessible problem, like the ones given in the table below:

The technique 'have I used everything' is perhaps most applicable to formal educational examinations (e.g., n men digging m ditches, see footnote below) problems. The book has achieved 'classic' status because of its considerable influence. Marvin Minsky[74] said in his influential paper 'Steps Toward Artificial Intelligence': "And everyone should know the work of George Polya

[74] Marvin Lee Minsky (born August 9, 1927), sometimes affectionately known as 'Old Man Minsky', is an American cognitive scientist in the field of artificial

on how to solve problems." Polya's book has had a large influence on mathematics textbooks. Most formulations of a problem solving framework in U.S. textbooks attribute some relationship to Polya's problem solving stages. Other books on problem solving are often related to less concrete and more creative techniques, like e.g., lateral thinking, mind mapping and brainstorming (see below).

On the other hand, in Europe, two main approaches have surfaced, one initiated by Donald Broadbent in the UK [Bro77, BB95] and the other one by Dietrich Dörner in Germany [Dor75, DV85, DW95]. The two approaches have in common an emphasis on relatively complex, semantically rich, computerized laboratory tasks, constructed to resemble 'real–life' problems. The approaches differ somewhat in their theoretical goals and methodology, however. The tradition initiated by Broadbent emphasizes the distinction between cognitive problem–solving processes that operate under awareness versus outside of awareness, and typically employs mathematically well–defined computerized systems. The tradition initiated by Dörner, on the other hand, has an interest in the interplay of the cognitive, motivational, and social components of problem solving, and utilizes very complex computerized scenarios that contain up to 2,000 highly interconnected variables. Buchner [Buc95] describes the two traditions in detail.

To sum up, researchers' realization that problem–solving processes differ across knowledge domains and across levels of expertise (see, e.g. [Ste95]) and that, consequently, findings obtained in the laboratory cannot necessarily generalize to problem–solving situations outside the laboratory, has during the past two decades led to an emphasis on real–world problem solving. This emphasis has been expressed quite differently in North America and Europe, however. Whereas North American research has typically concentrated on studying problem solving in separate, natural knowledge domains, much of the European research has focused on novel, complex problems, and has been performed with computerized scenarios (see [Fun95], for an overview).

Characteristics of Difficult Problems

As elucidated by Dietrich Dorner and later expanded upon by Joachim Funke, difficult problems have some typical characteristics. Recategorized and somewhat reformulated from these original works, these characteristics can be summarized as follows:

Intransparency (lack of clarity of the situation), including commencement opacity and continuation opacity;

Polytely (multiple goals), including inexpressiveness, opposition and transience;

---

intelligence (AI), co–founder of MIT's AI laboratory, and author of several texts on AI and philosophy.

Complexity (large numbers of items, interrelations, and decisions), including enumerability, connectivity (hierarchy relation, communication relation, allocation relation), and heterogeneity;

Dynamism (time considerations), including temporal constraints, temporal sensitivity, phase effects, and dynamic unpredictability.

The resolution of difficult problems requires a direct attack on each of these characteristics that are encountered.

Some *standard problem–solving techniques*, also known as creativity techniques, include:

1. Trial–and–error;[75]

[75] Trial and error (also known in computer science literature as generate and test and as 'guess and check' when solving equations in elementary algebra) is a method of problem solving for obtaining knowledge, both propositional knowledge and know-how.

This approach can be seen as one of the two basic approaches to problem solving and is contrasted with an approach using insight and theory.

In trial and error, one selects (or, generates) a possible answer, applies it to the problem and, if it is not successful, selects (or generates) another possibility that is subsequently tried. The process ends when a possibility yields a solution.

In some versions of trial and error, the option that is a priori viewed as the most likely one should be tried first, followed by the next most likely, and so on until a solution is found, or all the options are exhausted. In other versions, options are simply tried at random.

This approach is most successful with simple problems and in games, and is often resorted to when no apparent rule applies. This does not mean that the approach need be careless, for an individual can be methodical in manipulating the variables in an effort to sort through possibilities that may result in success. Nevertheless, this method is often used by people who have little knowledge in the problem area.

Trial and error has a number of features:

solution-oriented: trial and error makes no attempt to discover why a solution works, merely that it is a solution.

problem-specific: trial and error makes no attempt to generalize a solution to other problems.

non-optimal: trial and error is an attempt to find a solution, not all solutions, and not the best solution.

needs little knowledge: trial and error can proceed where there is little or no knowledge of the subject.

For example, trial and error has traditionally been the main method of finding new drugs, such as antibiotics. Chemists simply try chemicals at random until they find one with the desired effect.

The *scientific method* can be regarded as containing an element of trial and error in its formulation and testing of hypotheses. Also compare *genetic algorithms*, *simulated annealing* and *reinforcement learning* – all varieties of search which apply the basic idea of trial and error.

2. Brainstorming;[76]
3. Morphological box;[77]

Biological Evolution is also a form of trial and error. Random mutations and sexual genetic variations can be viewed as trials and poor reproductive fitness as the error. Thus after a long time 'knowledge' of well–adapted genomes accumulates simply by virtue of them being able to reproduce.

Bogosort can be viewed as a trial and error approach to sorting a list.

In mathematics the method of trial and error can be used to solve formulae – it is a slower, less precise method than algebra, but is easier to understand.

[76] Brainstorming is a creativity technique of generating ideas to solve a problem. The main result of a brainstorm session may be a complete solution to the problem, a list of ideas for an approach to a subsequent solution, or a list of ideas resulting in a plan to find a solution. Brainstorming was originated in 1953 in the book 'Applied Imagination' by Alex Osborn, an advertising executive. Other methods of generating ideas are individual ideation and the morphological analysis approach.

Brainstorming has many applications but it is most often used in:

New product development – obtaining ideas for new products and improving existing products

Advertising – developing ideas for advertising campaigns

Problem solving – issues, root causes, alternative solutions, impact analysis, evaluation

Process management – finding ways of improving business and production processes

Project Management – identifying client objectives, risks, deliverables, work packages, resources, roles and responsibilities, tasks, issues

Team building – generates sharing and discussion of ideas while stimulating participants to think

Business planning – develop and improve the product idea.

Trial preparation by attorneys.

Brainstorming can be done either individually or in a group. In group brainstorming, the participants are encouraged, and often expected, to share their ideas with one another as soon as they are generated. Complex problems or brainstorm sessions with a diversity of people may be prepared by a chairman. The chairman is the leader and facilitator of the brainstorm session.

The key to brainstorming is to not interrupt the thought process. As ideas come to mind, they are captured and stimulate the development of better ideas. Thus a group brainstorm session is best conducted in a moderate–sized room, and participants sit so that they can all look at each–other. A flip chart, blackboard, or overhead projector is placed in a prominent location. The room is free of telephones, clocks, or any other distractions.

[77] Morphological analysis was designed for multi-dimensional, non-quantifiable problems where causal modelling and simulation do not function well or at all. Fritz Zwicky developed this approach to seemingly non-reducible complexity [Zwi69]. Using the technique of cross consistency assessment (CCA) [Rit02], the system however does allow for reduction, not by reducing the number of variables involved, but by reducing the number of possible solutions through the elimination of the illogical solution combinations in a grid box.

4. Method of focal objects;[78]
5. Lateral thinking;[79]

[78] The technique of *focal objects* for problem solving involves synthesizing the seemingly non–matching characteristics of different objects into something new.

For example, to generate new solutions to gardening take some ideas at random, such swimming and a couch, and invent ways for them to merge. Swimming might be used with the idea of gardening to create a plant oxygen tank for underwater divers. A couch might be used with the idea of gardening to invent new genes that would grow plants into the shape of a couch. The larger the number of diverse objects included, the greater the opportunity for inventive solutions.

Another way to think of focal objects is as a memory cue: if you're trying to find all the different ways to use a brick, give yourself some random 'objects' (situations, concepts, etc.) and see if you can find a use. Given 'blender', for example, I would try to think of all the ways a brick could be used with a blender (as a lid?). Another concept for the brick game: find patterns in your solutions, and then break those patterns. If you keep finding ways to build things with bricks, think of ways to use bricks that don't involve construction. Pattern–breaking, combined with focal object cues, can lead to very divergent solutions.

[79] Lateral thinking is a term coined by Edward de Bono [Bon73], a Maltese psychologist, physician, and writer, although it may have been an idea whose time was ready; the notion of lateral truth is discussed by Robert M. Pirsig in Zen and the Art of Motorcycle Maintenance. de Bono defines Lateral Thinking as methods of thinking concerned with changing concepts and perception. For example:

It took two hours for two men to dig a hole five feet deep. How deep would it have been if ten men had dug the hole for two hours?

The answer appears to be 25 feet deep. This answer assumes that the thinker has followed a simple mathematical relationship suggested by the description given, but we can generate some lateral thinking ideas about what affects the size of the hole which may lead to different answers:

A hole may need to be of a certain size or shape so digging might stop early at a required depth.

The deeper a hole is, the more effort is required to dig it, since waste soil needs to be lifted higher to the ground level. There is a limit to how deep a hole can be dug by manpower without use of ladders or hoists for soil removal, and 25 feet is beyond this limit.

Deeper soil layers may be harder to dig out, or we may hit bedrock or the water table.

Each man digging needs space to use a shovel.

It is possible that with more people working on a project, each person may become less efficient due to increased opportunity for distraction, the assumption he can slack off, more people to talk to, etc.

More men could work in shifts to dig faster for longer.

There are more men but are there more shovels?

The two hours dug by ten men may be under different weather conditions than the two hours dug by two men.

Rain could flood the hole to prevent digging.

Temperature conditions may freeze the men before they finish.

Would we rather have 5 holes each 5 feet deep?

6. Mind mapping;[80]

The two men may be an engineering crew with digging machinery.

What if one man in each group is a manager who will not actually dig?

The extra eight men might not be strong enough to dig, or much stronger than the first two.

The most useful ideas listed above are outside the simple mathematics implied by the question. Lateral thinking is about reasoning that is not immediately obvious and about ideas that may not be obtainable by using only traditional step–by–step logic.

Techniques that apply lateral thinking to problems are characterized by the shifting of thinking patterns away from entrenched or predictable thinking to new or unexpected ideas. A new idea that is the result of lateral thinking is not always a helpful one, but when a good idea is discovered in this way it is usually obvious in hindsight, which is a feature lateral thinking shares with a joke.

Lateral thinking can be contrasted with critical thinking, which is primarily concerned with judging the truth value of statements and seeking error. Lateral Thinking is more concerned with the movement value of statements and ideas, how to move from them to other statements and ideas.

For example the statement 'cars should have square wheels' when considered with critical thinking would be evaluated as a poor suggestion, as there are many engineering problems with square wheels. The Lateral Thinking treatment of the same statement would be to see where it leads. Square wheels would produce predictable bumps. If bumps can be predicted then suspension can be designed to compensate. Another way to predict bumps would be a laser or sonar on the front of the car examining the road surface ahead. This leads to the idea of active suspension with a sensor on the car that has normal wheels. The initial statement has been left behind.

[80] Recall that a *mind map* is a diagram used to represent words, ideas, tasks or other items linked to and arranged radially around a central key word or idea. It is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, and decision making.

It is an image–centered diagram that represents semantic or other connections between portions of information. By presenting these connections in a radial, nonlinear graphical manner, it encourages a brainstorming approach to any given organizational task, eliminating the hurdle of initially establishing an intrinsically appropriate or relevant conceptual framework to work within.

A mind map is similar to a semantic network or cognitive map but there are no formal restrictions on the kinds of links used.

Most often the map involves images, words, and lines. The elements are arranged intuitively according to the importance of the concepts and they are organized into groupings, branches, or areas. The uniform graphic formulation of the semantic structure of information on the method of gathering knowledge, may aid recall of existing memories.

People have been using image centered radial graphic organization techniques referred to variably as mental or generic mind maps for centuries in areas such as engineering, psychology, and education, although the claim to the origin of the mind map has been made by a British popular psychology author, Tony Buzan.

7. Analogy with similar problems;[81] and

The mind map continues to be used in various forms, and for various applications including learning and education (where it is often taught as 'Webs' or 'Webbing'), planning and in engineering diagramming.

When compared with the earlier original concept map (which was developed by learning experts in the 1960s) the structure of a mind map is a similar, but simplified, radial by having one central key word.

Mind maps have many applications in personal, family, educational, and business situations, including note-taking, brainstorming (wherein ideas are inserted into the map radially around the center node, without the implicit prioritization that comes from hierarchy or sequential arrangements, and wherein grouping and organizing is reserved for later stages), summarizing, revising and general clarifying of thoughts. For example, one could listen to a lecture and take down notes using mind maps for the most important points or keywords. One can also use mind maps as a mnemonic technique or to sort out a complicated idea. Mind maps are also promoted as a way to collaborate in color pen creativity sessions.

[81] Recall that analogy is either the cognitive process of transferring information from a particular subject (the analogue or source) to another particular subject (the target), or a linguistic expression corresponding to such a process. In a narrower sense, analogy is an inference or an argument from a particular to another particular, as opposed to deduction, induction, and abduction, where at least one of the premises or the conclusion is general. The word analogy can also refer to the relation between the source and the target themselves, which is often, though not necessarily, a similarity, as in the biological notion of analogy.

Niels Bohr's model of the atom made an analogy between the atom and the solar system.Analogy plays a significant role in problem solving, decision making, perception, memory, creativity, emotion, explanation and communication. It lies behind basic tasks such as the identification of places, objects and people, for example, in face perception and facial recognition systems. It has been argued that analogy is 'the core of cognition'. Specifically analogical language comprises exemplification, comparisons, metaphors, similes, allegories, and parables, but not metonymy. Phrases like and so on, and the like, as if, and the very word like also rely on an analogical understanding by the receiver of a message including them. Analogy is important not only in ordinary language and common sense, where proverbs and idioms give many examples of its application, but also in science, philosophy and the humanities. The concepts of association, comparison, correspondence, homomorphism, iconicity, isomorphism, mathematical homology, metaphor, morphological homology, resemblance, and similarity are closely related to analogy. In cognitive linguistics, the notion of conceptual metaphor may be equivalent to that of analogy.

Analogy has been studied and discussed since classical antiquity by philosophers, scientists and lawyers. The last few decades have shown a renewed interest in analogy, most notable in cognitive science.

With respect to the terms source and target, there are two distinct traditions of usage:

The logical and mathematical tradition speaks of an arrow, homomorphism, mapping, or morphism from what is typically the more complex domain or source

8. Research;[82]

1.1.3 Human Mind

Recall that the word *mind* commonly refers to the collective aspects of *intellect* and *consciousness* which are manifest in some combination of *thought*, *perception*, *emotion*, *will*, *memory*, and *imagination*.

There are many theories of what the mind is and how it works, dating back to Plato, Aristotle and other Ancient Greek philosophers. Modern theories, based on a scientific understanding of the brain, see the mind as a phenomenon of psychology, and the term is often used more or less synonymously with *consciousness*.

The question of which human attributes make up the mind is also much debated. Some argue that only the 'higher' intellectual functions constitute

---

to what is typically the less complex codomain or target, using all of these words in the sense of mathematical category theory.

The tradition that appears to be more common in cognitive psychology, literary theory, and specializations within philosophy outside of logic, speaks of a mapping from what is typically the more familiar area of experience, the source, to what is typically the more problematic area of experience, the target.

[82] Research is often described as an active, diligent, and systematic process of inquiry aimed at discovering, interpreting, and revising facts. This intellectual investigation produces a greater understanding of events, behaviors, or theories, and makes practical applications through laws and theories. The term research is also used to describe a collection of information about a particular subject, and is usually associated with science and the scientific method.

The word research derives from Middle French; its literal meaning is 'to investigate thoroughly'.

Thomas Kuhn, in his book 'The Structure of Scientific Revolutions', traces an interesting history and analysis of the enterprize of research.

Basic research (also called fundamental or pure research) has as its primary objective the advancement of knowledge and the theoretical understanding of the relations among variables. It is exploratory and often driven by the researcher's curiosity, interest, or hunch. It is conducted without any practical end in mind, although it may have unexpected results pointing to practical applications. The terms "basic" or "fundamental" indicate that, through theory generation, basic research provides the foundation for further, sometimes applied research. As there is no guarantee of short-term practical gain, researchers often find it difficult to get funding for basic research. Research is a subset of invention.

Applied research is done to solve specific, practical questions; its primary aim is not to gain knowledge for its own sake. It can be exploratory, but is usually descriptive. It is almost always done on the basis of basic research. Applied research can be carried out by academic or industrial institutions. Often, an academic institution such as a university will have a specific applied research program funded by an industrial partner interested in that program. Common areas of applied research include electronics, informatics, computer science, material science, process engineering, drug design ...

mind: particularly reason and memory. In this view the emotions – love, hate, fear, joy – are more 'primitive' or subjective in nature and should be seen as different in nature or origin to the mind. Others argue that the rational and the emotional sides of the human person cannot be separated, that they are of the same nature and origin, and that they should all be considered as part of the individual mind.

In popular usage *mind* is frequently synonymous with *thought*: It is that private conversation with ourselves that we carry on 'inside our heads' during every waking moment of our lives. Thus we 'make up our minds,' or 'change our minds' or are 'of two minds' about something. One of the key attributes of the mind in this sense is that it is a private sphere. No–one else can 'know our mind.' They can only know what we communicate.

Both philosophers and psychologists remain divided about the nature of the mind. Some take what is known as the substantial view, and argue that the mind is a single entity, perhaps having its base in the brain but distinct from it and having an autonomous existence. This view ultimately derives from Plato, and was absorbed from him into Christian thought. In its most extreme form, the substantial view merges with the theological view that the mind is an entity wholly separate from the body, in fact a manifestation of the soul, which will survive the body's death and return to God, its creator.

Others take what is known as the functional view, ultimately derived from Aristotle, which holds that the mind is a term of convenience for a variety of mental functions which have little in common except that humans are conscious of their existence. Functionalists tend to argue that the attributes which we collectively call the mind are closely related to the functions of the brain and can have no autonomous existence beyond the brain, nor can they survive its death. In this view mind is a subjective manifestation of consciousness: the human brain's ability to be aware of its own existence. The concept of the mind is therefore a means by which the conscious brain understands its own operations.

A leading exponent of the *substantial view* at the mind was George Berkeley, an 18th century Anglican bishop and philosopher. Berkeley argued that there is no such thing as matter and what humans see as the material world is nothing but an idea in God's mind, and that therefore the human mind is purely a manifestation of the soul or spirit. This type of belief is also common in certain types of spiritual non–dualistic belief, but outside this field few philosophers take an extreme view today. However, the view that the human mind is of a nature or essence somehow different from, and higher than, the mere operations of the brain, continues to be widely held.

Berkeley's views were attacked, and in the eyes of many philosophers demolished, by T.H. Huxley,[83] a 19th century biologist and disciple of Charles

[83] Thomas Henry Huxley, FRS (4 May 1825 – 29 June 1895) was an English biologist, known as 'Darwin's Bulldog' for his defence of Charles Darwin's theory of evolution. His scientific debates against Richard Owen demonstrated that there were

Darwin,[84] who agreed that the phenomena of the mind were of a unique order, but argued that they can only be explained in reference to events in the brain. Huxley drew on a tradition of materialist thought in British philosophy dating to Thomas Hobbes,[85] who argued in the 17th century that mental events were ultimately physical in nature, although with the biological knowledge of his day he could not say what their physical basis was. Huxley blended Hobbes with Darwin to produce the modern *functional view*. Huxley's view was reinforced by the steady expansion of knowledge about the functions of the human brain. In the 19th century it was not possible to say with certainty how the brain carried out such functions as memory, emotion, perception and reason. This left the field open for substantialists to argue for an autonomous mind, or for a metaphysical theory of the mind. But each advance in the study of the brain during the 20th century made this harder, since it became more and more apparent that all the components of the mind have their origins in

close similarities between the cerebral anatomy of humans and gorillas. Huxley did not accept many of Darwin's ideas, such as gradualism and was more interested in advocating a materialist professional science than in defending natural selection.

A talented populariser of science, he coined the term 'agnosticism' to describe his stance on religious belief. He is credited with inventing the concept of 'biogenesis', a theory stating that all cells arise from other cells and also 'abiogenesis', describing the generation of life from non–living matter.

[84] Charles Robert Darwin (12 February 1809 – 19 April 1882) was an English naturalist who achieved lasting fame by producing considerable evidence that species originated through evolutionary change, at the same time proposing the scientific theory that natural selection is the mechanism by which such change occurs. This theory is now considered a cornerstone of biology.

Darwin developed an interest in natural history while studying first medicine, then theology, at university. Darwin's observations on his five-year voyage on the Beagle brought him eminence as a geologist and fame as a popular author. His biological finds led him to study the transmutation of species and in 1838 he conceived his theory of natural selection. Fully aware that others had been severely punished for such 'heretical' ideas, he confided only in his closest friends and continued his research to meet anticipated objections. However, in 1858 the information that Alfred Wallace had developed a similar theory forced an early joint publication of the theory.

His 1859 book 'On the Origin of Species by Means of Natural Selection' established evolution by common descent as the dominant scientific explanation of diversification in nature.

[85] Thomas Hobbes (April 5, 1588–December 4, 1679) was an English philosopher, whose famous 1651 book Leviathan set the agenda for nearly all subsequent Western political philosophy. Although Hobbes is today best remembered for his work on *political philosophy*, he contributed to a diverse array of fields, including history, geometry, ethics, general philosophy and what would now be called political science. Additionally, Hobbes's account of human nature as self–interested cooperation has proved to be an enduring theory in the field of philosophical anthropology.

the functioning of the brain. Huxley's rationalism, was disturbed in the early 20th century by Freudian a theory of the unconscious mind, and argued that those mental processes of which humans are subjectively aware are only a small part of their total mental activity.

More recently, Douglas Hofstadter's[86] 1979 Pulitzer Prize–winning book 'Gödel, Escher, Bach – an eternal Golden Braid', is a *tour de force* on the subject of mind, and how it might arise from the neurology of the brain. Amongst other biological and cybernetic phenomena, Hofstadter places tangled loops and recursion at the center of self, self–awareness, and perception of oneself, and thus at the heart of mind and thinking. Likewise philosopher Ken Wilber posits that Mind is the interior dimension of the brain holon, i.e., mind is what a brain looks like internally, when it looks at itself.

Quantum physicist David Bohm[87] had a theory of mind that is most comparable to Neo–Platonic theories. "Thought runs you. Thought, however, gives false info that you are running it, that you are the one who controls thought. Whereas actually thought is the one which controls each one of us ..." [Boh92].

The debate about the nature of the mind is relevant to the development of artificial intelligence (see next section). If the mind is indeed a thing separate from or higher than the functioning of the brain, then presumably it will not be possible for any machine, no matter how sophisticated, to duplicate it. If on the other hand the mind is no more than the aggregated functions of the

[86] Douglas Richard Hofstadter (born February 15, 1945 in New York, New York) is an American academic, the son of Nobel Prize–winning physicist Robert Hofstadter. He is probably best known for his book Gödel, Escher, Bach: an Eternal Golden Braid (abbreviated as GEB) which was published in 1979, and won the 1980 Pulitzer Prize for general non-fiction. This book is commonly considered to have inspired many students to begin careers in computing and artificial intelligence, and attracted substantial notice outside its central artificial intelligence readership owing to its drawing on themes from such diverse disciplines as high-energy physics, music, the visual arts, molecular biology, and literature.

[87] David Joseph Bohm (born December 20, 1917 in Wilkes–Barre, Pennsylvania, died October 27, 1992 in London) was an American–born quantum physicist, who made significant contributions in the fields of theoretical physics, philosophy and neuropsychology, and to the Manhattan Project.

Bohm made a number of significant contributions to physics, particularly in the area of quantum mechanics and relativity theory. While still a post-graduate at Berkeley, he developed a theory of plasmas, discovering the electron phenomenon now known as Bohm–diffusion. His first book, Quantum Theory published in 1951, was well–received by Einstein, among others. However, Bohm became dissatisfied with the orthodox approach to quantum theory, which he had written about in that book, and began to develop his own approach (Bohm interpretation), a non–local hidden variable deterministic theory whose predictions agree perfectly with the nondeterministic quantum theory. His work and the EPR argument became the major factor motivating John Bell's inequality, whose consequences are still being investigated.

brain, then it will be possible, at least in theory, to create a machine with a mind.

Currently, the Mind/Brain/Behavior Interfaculty Initiative (MBB) at Harvard University aims to elucidate the structure, function, evolution, development, and pathology of the nervous system in relation to human behavior and mental life. It draws on the departments of psychology, neurobiology, neurology, molecular and cellular biology, radiology, psychiatry, organismic and evolutionary biology, history of science, and linguistics.

Bohm also made significant theoretical contributions to neuropsychology and the development of the so–called *holonomic brain model*. In collaboration with Stanford neuroscientist Karl Pribram, Bohm helped establish the foundation for Pribram's theory that the brain operates in a manner similar to a hologram, in accordance with quantum mathematical principles and the characteristics of wave patterns. These wave forms may compose hologram–like organizations, Bohm suggested, basing this concept on his application of *Fourier analysis*, a mathematical method for decomposing complex waves into component sine waves. The holonomic brain model developed by Pribram and Bohm posits a lens defined world view, much like the textured prismatic effect of sunlight refracted by the churning mists of a rainbow, a view which is quite different from the more conventional 'objective' approach. Pribram believes that if psychology means to understand the conditions that produce the world of appearances, it must look to the thinking of physicists like Bohm.

Bohm proposes thus in his book 'Thought as a System' a pervasive, systematic nature of thought: "What I mean by 'thought' is the whole thing – thought, 'felt', the body, the whole society sharing thoughts – it's all one process. It is essential for me not to break that up, because it's all one process; somebody else's thoughts becomes my thoughts, and vice versa. Therefore it would be wrong and misleading to break it up into my thoughts, your thoughts, my feelings, these feelings, those feelings ... I would say that thought makes what is often called in modern language a system. A system means a set of connected things or parts. But the way people commonly use the word nowadays it means something all of whose parts are mutually interdependent – not only for their mutual action, but for their meaning and for their existence. A corporation is organized as a system – it has this department, that department, that department. They do not have any meaning separately; they only can function together. And also the body is a system. Society is a system in some sense. And so on. Similarly, thought is a system. That system not only includes thoughts and feelings, but it includes the state of the body; it includes the whole of society – as thought is passing back and forth between people in a process by which thought evolved from ancient times. A system is constantly engaged in a process of development, change, evolution and structure changes ... although there are certain features of the system which become relatively fixed. We call this the structure ... Thought has been constantly evolving and we can't say when that structure began. But with the growth of civilization it has developed a great deal. It was probably very simple thought before civilization, and now it has become very complex and ramified and has much more incoherence than before ...

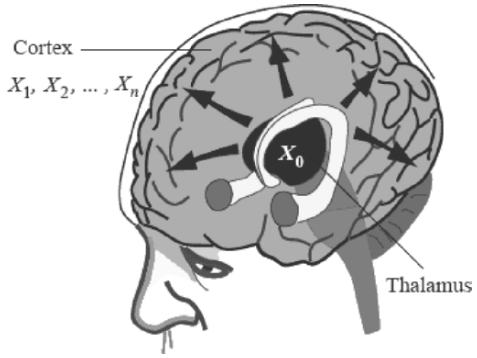

Fig. 1.2. A *possibly chaotic* 1–to–many relation: *Thalamus* $\Rightarrow$ *Cortex* in the human brain (with permission from E. Izhikevich).

On the other hand, human brain has been considered (by E.M. Izhikevich, Editor of the new Encyclopedia of Computational Neuroscience) as a *weakly–connected neural network*, with possibly *chaotic behavior* [Izh99b], consisting of n quasi–periodic cortical oscillators $X_1, \ldots, X_n$ forced by the thalamic input X_0 (see Figure 1.2)

The Mind–Body Problem

The *mind–body problem* is essentially the problem of explaining the relationship between minds, or mental processes, and bodily states or processes (see, e.g., [Kim95a]). Our perceptual experiences depend on stimuli which arrive at our various sensory organs from the external world and that these stimuli cause changes in the states of our brain, ultimately causing us to feel a sensation which may be pleasant or unpleasant. Someone's desire for a slice of pizza will tend to cause that person to move their body in a certain manner in a certain direction in an effort to get what they want. But how is it possible that conscious experiences can arise out of an inert lump of gray matter endowed with electrochemical properties? [Kim95b]. How does someone's desire cause that individual's neurons to fire and his muscles to contract in exactly the right manner? These are some of the essential puzzles that have confronted philosophers of mind at least from the time of René Descartes.[88]

[88] René Descartes (March 31, 1596 – February 11, 1650), also known as Cartesius, was a noted French philosopher, mathematician, and scientist. Dubbed the 'Founder of Modern Philosophy' and the 'Father of Modern Mathematics', he ranks as one of the most important and influential thinkers of modern times. Much of subsequent western philosophy is a reaction to his writings, which have been closely studied from his time down to the present day. Descartes was one of the key thinkers of the Scientific Revolution in the Western World. He is also

Dualism

Recall that *dualism* is a set of views about the relationship between mind and matter, which begins with the claim that mental phenomena are, in some respects, non–physical [Har96]. One of the earliest known formulations of mind–body dualism existed in the eastern *Sankhya school* of Hindu philosophy (c. 650 BCE) which divided the world into *Purusha* (mind/spirit) and *Prakrti* (material substance). In the Western philosophical tradition, we first encounter similar ideas with the writings of Plato and Aristotle, who maintained, for different reasons, that man's *intelligence* could not be identified with, or explained in terms of, his physical body (see, e.g., [RPW97]). However, the best–known version of dualism is due to René Descartes (1641), and holds that the mind is a non–physical substance [Des91]. Descartes was the first to clearly identify the mind with consciousness and self–awareness and to distinguish this from the brain, which was the seat of intelligence. Hence, he was the first to formulate the mind–body problem in the form in which it still exists today.

The main argument in favour of dualism is simply that it appeals to the common–sense intuition of the vast majority of non–philosophically–trained people. If asked what the mind is, the average person will usually respond by identifying it with their self, their personality, their soul, or some other such entity, and they will almost certainly deny that the mind simply is the brain or vice–versa, finding the idea that there is just one ontological entity at play to be too mechanistic or simply unintelligible [Har96]. The majority of modern philosophers of mind reject dualism, suggesting that these intuitions, like many others, are probably misleading. We should use our critical faculties, as well as empirical evidence from the sciences, to examine these assumptions and determine if there is any real basis to them [Har96] Another very important, more modern, argument in favor of dualism consists in the idea that the mental and the physical seem to have quite different and perhaps irreconcilable properties [Jac82]. Mental events have a certain subjective quality to them, whereas physical events obviously do not. For example, what does a burned finger feel like? What does blue sky look like? What does nice

honoured by having the *Cartesian coordinate system* used in plane geometry and algebra named after him.

Descartes was a major figure in 17th century continental rationalism, later advocated by Baruch Spinoza and Gottfried Leibniz, and opposed by the empiricist school of thought, consisting of Hobbes, Locke, Berkeley, and Hume. Leibniz, Spinoza and Descartes were all versed in mathematics as well as philosophy, and Descartes and Leibniz contributed greatly to science as well. As the inventor of the Cartesian coordinate system, Descartes founded analytic geometry, that bridge between algebra and geometry crucial to the invention of the calculus and analysis. Descartes' reflections on mind and mechanism began the strain of western thought that much later, impelled by the invention of the electronic computer and by the possibility of machine intelligence, blossomed into, e.g., the Turing test. His most famous statement is "Cogito ergo sum" (I think, therefore I am).

music sound like? Philosophers of mind call the subjective aspects of mental events qualia (or raw feels) [Jac82]. There is something that it is like to feel pain, to see a familiar shade of blue, and so on; there are qualia involved in these mental events. And the claim is that qualia seem particularly difficult to reduce to anything physical [Nag74].

Interactionist dualism, or simply *interactionism*, is the particular form of dualism first espoused by Descartes in the 'Meditations' [Des91]. In the 20th century, its major defenders have been Karl Popper[89] and John Eccles[90]

[89] Sir Karl Raimund Popper (July 28, 1902 – September 17, 1994), was an Austrian and British philosopher and a professor at the London School of Economics. He is counted among the most influential philosophers of science of the 20th century, and also wrote extensively on social and political philosophy. Popper is perhaps best known for repudiating the classical observationalist–inductivist account of scientific method by advancing empirical falsifiability as the criterion for distinguishing scientific theory from non–science; and for his vigorous defense of liberal democracy and the principles of social criticism which he took to make the flourishing of the 'open society' possible. In 1934 he published his first book, 'The Logic of Scientific Discovery', in which he criticized psychologism, naturalism, inductionism, and logical positivism, and put forth his theory of potential falsifiability being the criterion for what should be considered science.

Popper coined the term *critical rationalism* to describe his philosophy. This designation is significant, and indicates his rejection of classical empiricism, and of the observationalist-inductivist account of science that had grown out of it. Popper argued strongly against the latter, holding that scientific theories are universal in nature, and can be tested only indirectly, by reference to their implications. He also held that scientific theory, and human knowledge generally, is irreducibly conjectural or hypothetical, and is generated by the creative imagination in order to solve problems that have arisen in specific historico–cultural settings. Logically, no number of positive outcomes at the level of experimental testing can confirm a scientific theory, but a single genuine counterexample is logically decisive: it shows the theory, from which the implication is derived, to be false. Popper's account of the logical asymmetry between verification and falsification lies at the heart of his philosophy of science. It also inspired him to take falsifiability as his criterion of demarcation between what is and is not genuinely scientific: a theory should be considered scientific if and only if it is falsifiable. This led him to attack the claims of both psychoanalysis and contemporary Marxism to scientific status, on the basis that the theories enshrined by them are not falsifiable. His scientific work was influenced by his study of quantum mechanics (he has written extensively against the famous Copenhagen interpretation) and by Albert Einstein's approach to scientific theories.

In his book 'All Life is Problem Solving' (1999), Popper sought to explain the apparent progress of scientific knowledge, how it is that our understanding of the universe seems to improve over time. This problem arises from his position that the truth content of our theories, even the best of them, cannot be verified by scientific testing, but can only be falsified. If so, then how is it that the growth

(see [PE02]). It is the view that mental states, such as beliefs and desires, causally interact with physical states [Har96]. Descartes' famous argument for this position can be summarized as follows: Fred has a clear and distinct idea of his mind as a thinking thing which has no spatial extension (i.e., it cannot be measured in terms of length, weight, height, and so on) and he also has a clear and distinct idea of his body as something that is spatially

---

of science appears to result in a growth in knowledge? In Popper's view, the advance of scientific knowledge is an evolutionary process characterised by his formula:

$$PS_1 \rightarrow TT_1 \rightarrow EE_1 \rightarrow PS_2\,.$$

In response to a given problem situation, $PS1$, a number of competing conjectures, or tentative theories, TT, are systematically subjected to the most rigorous attempts at falsification possible. This process, error elimination, EE, performs a similar function for science that natural selection performs for biological evolution. Theories that better survive the process of refutation are not more true, but rather, more 'fit', in other words, more applicable to the problem situation at hand, $PS1$. Consequently, just as a species' 'biological fit' does not predict continued survival, neither does rigorous testing protect a scientific theory from refutation in the future. Yet, as it appears that the engine of biological evolution has produced, over time, adaptive traits equipped to deal with more and more complex problems of survival, likewise, the evolution of theories through the scientific method may, in Popper's view, reflect a certain type of progress: toward more and more interesting problems, $PS2$. For Popper, it is in the interplay between the tentative theories (conjectures) and error elimination (refutation) that scientific knowledge advances toward greater and greater problems; in a process very much akin to the interplay between genetic variation and natural selection.

As early as 1934 Popper wrote of the *search for truth* as one of the "strongest motives for scientific discovery." Still, he describes in 'Objective Knowledge' (1972) early concerns about the much–criticised notion of *truth as correspondence*. Then came the *semantic theory of truth* formulated by the logician Alfred Tarski. Popper writes of learning in 1935 of the consequences of Tarski's theory, to his intense joy. The theory met critical objections to truth as correspondence and thereby rehabilitated it. The theory also seemed to Popper to support metaphysical realism and the regulative idea of a search for truth.

Among his contributions to philosophy is his answer to David Hume's 'Problem of Induction'. Hume stated that just because the sun has risen every day for as long as anyone can remember, doesn't mean that there is any rational reason to believe it will come up tomorrow. There is no rational way to prove that a pattern will continue on just because it has before. Popper's reply is characteristic, and ties in with his *criterion of falsifiability*. He states that while there is no way to prove that the sun will come up, we can theorize that it will. If it does not come up, then it will be disproven, but since right now it seems to be consistent with our theory, the theory is not disproven. Thus, Popper's demarcation between science and non–science serves as an answer to an old logical problem as well. This approach was criticised by Peter Singer for masking the role induction plays in empirical discovery.

extended, subject to quantification and not able to think. It follows that mind and body are not identical because they have radically different properties, according to Descartes [Des91]. At the same time, however, it is clear that Fred's mental states (desires, beliefs, etc.) have causal effects on his body and vice-versa: a child touches a hot stove (physical event) which causes pain (mental event) and makes him yell (physical event) which provokes a sense of fear and protectiveness in the mother (mental event) and so on. Descartes' argument obviously depends on the crucial premise that what Fred believes to be 'clear and distinct' ideas in his mind are necessarily true. Most modern philosophers doubt the validity of such an assumption, since it has been shown

[90] Sir John Carew Eccles (January 27, 1903 – May 2, 1997) was an Australian neurophysiologist who won the 1963 Nobel Prize in Physiology or Medicine for his work on the synapse. He shared the prize together with Andrew Fielding Huxley and Alan Lloyd Hodgkin.

In the early 1950s, Eccles and his colleagues performed the key experiments that would win Eccles the Nobel Prize. To study synapses in the peripheral nervous system, Eccles and colleagues used the stretch reflex as a model. This reflex is easily studied because it consists of only two neurons: a sensory neuron (the muscle spindle fiber) and the motor neuron. The sensory neuron synapses onto the motor neuron in the spinal cord. When Eccles passed a current into the sensory neuron in the quadriceps, the motor neuron innervating the quadriceps produced a small excitatory postsynaptic potential (EPSP). When he passed the same current through the hamstring, the opposing muscle to the quadriceps, he saw an inhibitory postsynaptic potential (IPSP) in the quadriceps motor neuron. Although a single EPSP was not enough to fire an action potential in the motor neuron, the sum of several EPSPs from multiple sensory neurons synapsing onto the motor neuron could cause the motor neuron to fire, thus contracting the quadriceps. On the other hand, IPSPs could subtract from this sum of EPSPs, preventing the motor neuron from firing.

Apart from these seminal experiments, Eccles was key to a number of important developments in neuroscience. Until around 1949, Eccles believed that synaptic transmission was primarily electrical rather than chemical. Although he was wrong in this hypothesis, his arguments led himself and others to perform some of the experiments which proved chemical synaptic transmission. Bernard Katz and Eccles worked together on some of the experiments which elucidated the role of acetylcholine as a neurotransmitter.

[91] Pierre Maurice Marie Duhem (10 June 1861 – 14 September 1916) French physicist and philosopher of science. Duhem's sophisticated views on the philosophy of science are explicated in 'The aim and structure of physical theory' (foreword by Prince Louis de Broglie). In this work he refuted the inductivist untruth that Newton's laws can de deduced from Kepler, *et al.* (a selection was published as Medieval cosmology: theories of infinity, place, time, void, and the plurality of worlds. He gave his name to the Quine-Duhem thesis, which holds that for any given set of observations there are an innumerably large number of explanations. Thus empirical evidence cannot force the revision of a theory.

in modern times by Freud (a third–person psychologically–trained observer can understand a person's unconscious motivations better than she does), by Pierre Duhem[91]
(a third–person philosopher of science can know a person's methods of discovery better than she does), by Bronisław Malinowski[92] (an anthropologist can know a person's customs and habits better than he does), and by theorists of perception (experiments can make one see things that are not there and scientists can describe a person's perceptions better than he can), that such an idea of privileged and perfect access to one's own ideas is dubious at best.

Other important forms of dualism which arose as reactions to, or attempts to salvage, the Cartesian version are:

(i) Psycho–physical parallelism, or simply parallelism, is the view that mind and body, while having distinct ontological statuses, do not causally influence one another, but run along parallel paths (mind events causally interact with mind events and brain events causally interact with brain events) and only seem to influence each other [RPW97]. This view was most prominently defended by Gottfried Leibniz.[93] Although Leibniz was actually an ontological monist who believed that only one fundamental substance, monads, exists in the universe and everything else is reducible to it, he nonetheless maintained that there was an important distinction between 'the mental' and 'the physical' in terms of causation. He held that God had arranged things in advance so that minds and bodies would be in harmony with each other. This is known as the doctrine of pre–established harmony [Lei714].

---

[92] Bronisław Kasper Malinowski (April 7, 1884 – May 16, 1942) was a Polish anthropologist widely considered to be one of the most important anthropologists of the twentieth century because of his pioneering work on ethnographic fieldwork, the study of reciprocity, and his detailed contribution to the study of Melanesia.

[93] Gottfried Wilhelm Leibniz (July 1 (June 21 Old Style) 1646 – November 14, 1716) was a German polymath. Educated in law and philosophy, Leibniz played a major role in the European politics and diplomacy of his day. He occupies an equally large place in both the history of philosophy and the history of mathematics. He invented *calculus* independently of Newton, and his notation is the one in general use since. He also invented the *binary system*, foundation of virtually all modern computer architectures. In philosophy, he is most remembered for *optimism*, i.e., his conclusion that our universe is, in a restricted sense, the best possible one God could have made. He was, along with René Descartes and Baruch Spinoza, one of the three great 17th century rationalists, but his philosophy also both looks back to the *Scholastic tradition* and anticipates logic and analysis. Leibniz also made major contributions to physics and technology, and anticipated notions that surfaced much later in biology, medicine, geology, probability theory, psychology, knowledge engineering, and information science. He also wrote on politics, law, ethics, theology, history, and philology, even occasional verse. His contributions to this vast array of subjects are scattered in journals and in tens of thousands of letters and unpublished manuscripts. To date, there is no complete edition of Leibniz's writings, and a complete account of his accomplishments is not yet possible.

(ii) Occasionalism is the view espoused by Nicholas Malebranche which asserts that all supposedly causal relations between physical events or between physical and mental events are not really causal at all. While body and mind are still different substances on this view, causes (whether mental or physical) are related to their effects by an act of God's intervention on each specific occasion [Sch02].

(iii) Epiphenomenalism is a doctrine first formulated by Thomas Huxley [Hux898]. Fundamentally, it consists in the view that mental phenomena are causally inefficacious. Physical events can cause other physical events and physical events can cause mental events, but mental events cannot cause anything, since they are just causally inert by-products (i.e. epiphenomena) of the physical world [RPW97] The view has been defended most strongly in recent times by Frank Jackson [Jac82].

(iv) Property dualism asserts that when matter is organized in the appropriate way (i.e., in the way that living human bodies are organized), mental properties emerge. Hence, it is a sub–branch of emergent materialism [Har96]. These emergent properties have an independent ontological status and cannot be reduced to, or explained in terms of, the physical substrate from which they emerge. This position is espoused by David Chalmers and has undergone something of a renaissance in recent years [Cha97].

Monism

In contrast to dualism, *monism* states that there is only one fundamental substance. Monism, first proposed in the West by Parmenides[94] and in modern times by Baruch Spinoza,[95] maintains that there is only one substance; in the East, rough parallels might be the Hindu concept of *Brahman* or the *Tao* of Lao Tzu [Spi670]. Today the most common forms of monism in Western philosophy are physicalistic [Kim95b]. Physicalistic monism asserts that the only existing substance is physical, in some sense of that term to be clarified

[94] Parmenides of Elea (early 5th century BC) was an ancient Greek philosopher born in Elea, a Hellenic city on the southern coast of Italy. Parmenides was a student of Ameinias and the founder of the School of Elea, which also included Zeno of Elea and Melissus of Samos.

[95] Benedictus de Spinoza (November 24, 1632 – February 21, 1677), named Baruch Spinoza by his synagogue elders, was a Jewish–Dutch philosopher. He is considered one of the great rationalists of 17th–century philosophy and, by virtue of his magnum opus the 'Ethics', one of the definitive ethicists. His writings, like those of his fellow rationalists, reveal considerable mathematical training and facility. Spinoza was a lens crafter by trade, an exciting engineering field at the time because of great discoveries being made by telescopes. The full impact of his work only took effect some time after his death and after the publication of his 'Opera Posthuma'. He is now seen as having prepared the way for the 18th century Enlightenment, and as a founder of modern biblical criticism. 20th century philosopher, Gilles Deleuze (1990), referred to Spinoza as "The absolute philosopher, whose Ethics is the foremost book on concepts."

by our best science [Sto05]. Another form of monism is that which states that the only existing substance is mental. Such idealistic monism is currently somewhat uncommon in the West [Kim95b].

Phenomenalism, the theory that all that exists are the representations (or sense data) of external objects in our minds and not the objects themselves, was adopted by Bertrand Russell[96] and many of the logical positivists during

[96] Bertrand Arthur William Russell, (3rd Earl Russell, 18 May 1872 – 2 February 1970), was a British philosopher, logician, and mathematician, working mostly in the 20th century. A prolific writer, Bertrand Russell was also a populariser of philosophy and a commentator on a large variety of topics, ranging from very serious issues to the mundane. Continuing a family tradition in political affairs, he was a prominent liberal as well as a socialist and anti–war activist for most of his long life. Millions looked up to Russell as a prophet of the creative and rational life; at the same time, his stances on many topics were extremely controversial.

Russell was born at the height of Britain's economic and political ascendancy. He died of influenza nearly a century later, at a time when the British Empire had all but vanished, its power dissipated by two debilitating world wars. As one of the world's best–known intellectuals, Russell's voice carried great moral authority, even into his early 90s. Among his political activities, Russell was a vigorous proponent of nuclear disarmament and an outspoken critic of the American war in Vietnam.

In 1950, Russell was made a Nobel Laureate in Literature, "in recognition of his varied and significant writings in which he champions humanitarian ideals and freedom of thought."

Russell is generally recognized as one of the founders of *analytical philosophy*, even of its several branches. At the beginning of the 20th century, alongside G.E. Moore, Russell was largely responsible for the British 'revolt against Idealism', a philosophy greatly influenced by Georg Hegel. This revolt was echoed 30 years later in Vienna by the logical positivists' 'revolt against metaphysics'. Russell was particularly appalled by the idealist doctrine of internal relations, which held that in order to know any particular thing, we must know all of its relations. Russell showed that this would make space, time, science and the concept of number unintelligible. Russell's logical work with Alfred Whitehead continued this project.

Russell had great influence on modern mathematical logic. His first mathematical book, An Essay on the Foundations of Geometry, was published in 1897. This work was heavily influenced by Immanuel Kant. Russell soon realised that the conception it laid out would have made Albert Einstein's schema of space-time impossible, which he understood to be superior to his own system. Thenceforth, he rejected the entire Kantian program as it related to mathematics and geometry, and he maintained that his own earliest work on the subject was nearly without value. Russell discovered that Gottlob Frege had independently arrived at equivalent definitions for 0, successor, and number, and the definition of number is now usually referred to as the *Frege–Russell definition*. It was largely Russell who brought Frege to the attention of the English-speaking world. He did this in 1903, when he published 'The Principles of Mathematics', in which the concept of class is inextricably tied to the definition of number. The appendix to this work detailed a paradox arising in Frege's application of second– and higher–order

the early 20th century [Rus18]. It lasted for only a very brief period of time. A third possibility is to accept the existence of a basic substance which is neither physical nor mental. The mental and physical would both be properties of this neutral substance. Such a position was adopted by Baruch Spinoza [Spi670] and popularized by Ernst Mach[97] [Mac59] in the 19th century. This neutral monism, as it is called, resembles property dualism.

Behaviorism

Behaviorism dominated philosophy of mind for much of the 20th century, especially the first half [Kim95b]. In psychology, *behaviorism* developed as a reaction to the inadequacies of introspectionism. Introspective reports on one's own interior mental life are not subject to careful examination for accuracy and are not generalizable. Without generalizability and the possibility of third-person examination, the behaviorists argued, science is simply not possible [Sto05]. The way out for psychology was to eliminate the idea of an interior mental life (and hence an ontologically independent mind) altogether and focus instead on the description of observable behavior [Ski72].

functions which took first-order functions as their arguments, and he offered his first effort to resolve what would henceforth come to be known as the *Russell Paradox*, which he later developed into a complete theory, the Theory of types. Aside from exposing a major inconsistency in naive set theory, Russell's work led directly to the creation of modern axiomatic set theory. It also crippled Frege's project of reducing arithmetic to logic. The Theory of Types and much of Russell's subsequent work have also found practical applications with computer science and information technology.

Russell continued to defend *logicism*, the view that mathematics is in some important sense reducible to logic, and along with his former teacher, Alfred Whitehead, wrote the monumental 'Principia Mathematica', an *axiomatic system* on which all of mathematics can be built. The first volume of the Principia was published in 1910, and is largely ascribed to Russell. More than any other single work, it established the specialty of mathematical or symbolic logic. Two more volumes were published, but their original plan to incorporate geometry in a fourth volume was never realised, and Russell never felt up to improving the original works, though he referenced new developments and problems in his preface to the second edition. Upon completing the Principia, three volumes of extraordinarily abstract and complex reasoning, Russell was exhausted, and he never felt his intellectual faculties fully recovered from the effort. Although the Principia did not fall prey to the paradoxes in Frege's approach, it was later proven by Kurt Gödel that neither Principia Mathematica, nor any other consistent system of primitive recursive arithmetic, could, within that system, determine that every proposition that could be formulated within that system was decidable, i.e., could decide whether that proposition or its negation was provable within the system (*Gödel's incompleteness theorem*).

[97] Ernst Mach (February 18, 1838 – February 19, 1916) was an Austrian–Czech physicist and philosopher and is the namesake for the 'Mach number" (aka Mach speed) and the optical illusion known as Mach bands.

Parallel to these developments in psychology, a philosophical behaviorism (sometimes called logical behaviorism) was developed [Sto05]. This is characterized by a strong verificationism, which generally considers unverifiable statements about interior mental life senseless. But what are mental states if they are not interior states on which one can make introspective reports? The answer of the behaviorist is that mental states do not exist but are actually just descriptions of behavior and/or dispositions to behave made by external third parties in order to explain and predict others' behavior [Ryl49]. Philosophical behaviorism is considered by most modern philosophers of mind to be outdated [Kim95a]. Apart from other problems, behaviorism implausibly maintains, for example, that someone is talking about behavior if she reports that she has a wracking headache.

Continental Philosophy of Mind

In contrast to Anglo–American *analytic philosophy*[98] there are other schools of thought which are sometimes subsumed under the broad label of *continental philosophy*. These schools tend to differ from the analytic school in

[98] Analytic philosophy is the dominant academic philosophical movement in English–speaking countries and in the Nordic countries. It is distinguished from Continental Philosophy which pertains to most non–English speaking countries. Its main founders were the Cambridge philosophers G.E. Moore and Bertrand Russell. However, both were heavily influenced by the German philosopher and mathematician Gottlob Frege and many of analytic philosophy's leading proponents, such as Ludwig Wittgenstein, Rudolf Carnap, Kurt Gödel, Karl Popper, Hans Reichenbach, Herbert Feigl, Otto Neurath, and Carl Hempel have come from Germany and Austria. In Britain, Russell and Moore were succeeded by C. D. Broad, L. Stebbing, Gilbert Ryle, A. J. Ayer, R. B. Braithwaite, Paul Grice, John Wisdom, R. M. Hare, J. L. Austin, P. F. Strawson, William Kneale, G. E. M. Anscombe, and Peter Geach. In America, the movement was led by many of the above-named European emigres as well as Max Black, Ernest Nagel, C. L. Stevenson, Norman Malcolm, W. V. Quine, Wilfrid Sellars, and Nelson Goodman, while A. N. Prior, John Passmore, and J. J. C. Smart were prominent in Australasia.

Logic and philosophy of language were central strands of analytic philosophy from the beginning, although this dominance has diminished greatly. Several lines of thought originate from the early, language-and-logic part of this analytic philosophy tradition. These include: logical positivism, logical empiricism, logical atomism, logicism and ordinary language philosophy. Subsequent analytic philosophy includes extensive work in ethics (such as Philippa Foot, R. M. Hare, and J. L. Mackie), political philosophy (John Rawls, Robert Nozick), aesthetics (Monroe Beardsley, Richard Wollheim, Arthur Danto), philosophy of religion (Alvin Plantinga, Richard Swinburne), philosophy of language (David Kaplan, Saul Kripke, Richard Montague, Hilary Putnam, W.V.O. Quine, Nathan Salmon, John Searle), and philosophy of mind (Daniel Dennett, David Chalmers, Putnam). Analytic metaphysics has also recently come into its own (Kripke, David Lewis, Salmon, Peter van Inwagen, P.F. Strawson).

that they focus less on language and logical analysis and more on directly understanding human existence and experience. With reference specifically to the discussion of the mind, this tends to translate into attempts to grasp the concepts of thought and perceptual experience in some direct sense that does not involve the analysis of linguistic forms [Dum01]. In particular, in his 'Phenomenology of Mind', G.W. F. Hegel[99] discusses three distinct types of mind: the subjective mind, the mind of an individual; the objective mind, the mind of society and of the State; and the Absolute mind, a unity of all concepts. In modern times, the two main schools that have developed in response or opposition to this Hegelian tradition are *phenomenology* and *existentialism*. Phenomenology, founded by Edmund Husserl,[100] focuses on the contents of

[99] Georg Wilhelm Friedrich Hegel (August 27, 1770 – November 14, 1831) was a German philosopher born in Stuttgart, Württemberg, in present-day southwest Germany. His influence has been widespread on writers of widely varying positions, including both his admirers (F.H. Bradley, J.P. Sartre, Hans Küng, Bruno Bauer), and his detractors (Kierkegaard, Schopenhauer, Heidegger, Schelling). His great achievement was to introduce for the first time in philosophy the idea that History and the concrete are important in getting out of the circle of philosophia perennis, i.e., the perennial problems of philosophy. Also, for the first time in the history of philosophy he realised the importance of the Other in the coming to be of self–consciousness, see slave–master dialectic.

Some of Hegel's writing was intended for those with advanced knowledge of philosophy, although his 'Encyclopedia' was intended as a textbook in a university course. Nevertheless, like many philosophers, Hegel assumed that his readers would be well–versed in Western philosophy, up to and including Descartes, Spinoza, Hume, Kant, Fichte, and Schelling. For those wishing to read his work without this background, introductions to Hegel and commentaries about Hegel may suffice. However, even this is hotly debated since the reader must choose from multiple interpretations of Hegel's writings from incompatible schools of philosophy. Presumably, reading Hegel directly would be the best method of understanding him, but this task has historically proved to be beyond the average reader of philosophy.[citation needed] This difficulty may be the most urgent problem with respect to the legacy of Hegel.

One especially difficult aspect of Hegel's work is his innovation in logic. In response to Immanuel Kant's challenge to the limits of Pure Reason, Hegel developed a radically new form of logic, which he called speculation, and which is today popularly called *dialectics*. The difficulty in reading Hegel was perceived in Hegel's own day, and persists into the 21st century. To understand Hegel fully requires paying attention to his critique of standard logic, such as the *law of contradiction* and the *law of the excluded middle*, and, whether one accepts or rejects it, at least taking it seriously. Many philosophers who came after Hegel and were influenced by him, whether adopting or rejecting his ideas, did so without fully absorbing his new speculative or dialectical logic.

[100] Edmund Gustav Albrecht Husserl (April 8, 1859, Prostějov – April 26, 1938, Freiburg) was a German philosopher, known as the father of phenomenology. Husserl was born into a Jewish family in Prostějov (Prossnitz), Moravia, Czech Republic (then part of the Austrian Empire). A pupil of Franz Brentano and

the human mind and how phenomenological processes shape our experiences. Existentialism, a school of thought led by Jean–Paul Sartre,[101] focuses on the content of experiences and how the mind deals with such experiences [Fly04].

Neurobiology

On the other hand, within the tangible field of *neurobiology*, there are many subdisciplines which are concerned with the relations between mental and physical states and processes [Bea95]:

1. Sensory neurophysiology investigates the relation between the processes of perception and stimulation [Pine97].

Carl Stumpf, Husserl came to influence, among others, Edith Stein (St. Teresa Benedicta of the Cross), Eugen Fink, Martin Heidegger, Jean–Paul Sartre, and Maurice Merleau–Ponty; in addition, Hermann Weyl's interest in *intuitionistic logic* and impredicativity appear to have resulted from contacts with Husserl. Rudolf Carnap was also influenced by Husserl, not only concerning Husserl's notion of essential insight that Carnap used in his Der Raum, but also his notion of *formation rules* and *transformation rules* is founded on Husserl's philosophy of logic. In 1887 Husserl converted to Christianity and joined the Lutheran Church. He taught philosophy at Halle as a tutor (Privatdozent) from 1887, then at Göttingen as professor from 1901, and at Freiburg im Breisgau from 1916 until he retired in 1928. After this, he continued his research and writing by using the library at Freiburg, until barred therefrom because of his Jewish heritage under the rectorship of his former pupil and intended protege, Martin Heidegger.

Husserl held the belief that *truth–in–itself* has as ontological correlate *being–in–itself*, just as meaning categories have formal–ontological categories as correlates. The discipline of logic is a formal theory of judgment, that studies the formal a priori relations among judgments using meaning categories. Mathematics, on the other hand, is formal ontology, it studies all the possible forms of being (of objects). So, in both of these disciplines, formal categories, in their different forms, are their object of study, not the sensible objects themselves. The problem with the psychological approach to mathematics and logic is that it fails to account for the fact that it is about formal categories, not abstractions from sensibility alone. The reason why we do not deal with sensible objects in mathematics is because of another faculty of understanding called *categorial abstraction*. Through this faculty we are able to get rid of sensible components of judgments, and just focus on formal categories themselves. Thanks to 'eidetic (or essential) intuition', we are able to grasp the possibility, impossibility, necessity and contingency among concepts or among formal categories. Categorial intuition, along with categorial abstraction and eidetic intuition, are the basis for logical and mathematical knowledge.

[101] Jean–Paul Charles Aymard Sartre (June 21, 1905 – April 15, 1980), was a French existentialist philosopher, dramatist and screenwriter, novelist and critic.

The basis of Sartre's existentialism is found in his 'The Transcendence of the Ego'. To begin with, the thing–in–itself is infinite and overflowing. Any direct consciousness of the thing–in–itself, Sartre refers to as a 'pre–reflective consciousness'. Any attempt to describe, understand, historicize etc. the thing–in–itself,

2. Cognitive neuroscience studies the correlations between mental processes and neural processes [Pine97].
3. Neuropsychology describes the dependence of mental faculties on specific anatomical regions of the brain [Pine97].
4. Lastly, evolutionary biology studies the origins and development of the human nervous system and, in as much as this is the basis of the mind, also describes the ontogenetic and phylogenetic development of mental phenomena beginning from their most primitive stages [Pink97].

Since the 1980's, sophisticated neuroimaging procedures, such as fMRI, have furnished increasing knowledge about the workings of the human brain, shedding light on ancient philosophical problems.The methodological breakthroughs of the neurosciences, in particular the introduction of high–tech neuroimaging procedures, has propelled scientists toward the elaboration of increasingly ambitious research programs: one of the main goals is to describe and comprehend the neural processes which correspond to mental functions [Bea95]. A very small number of neurobiologists, such as Emil Reymond[102] be and John Eccles have denied the possibility of a 'reduction' of mental phenomena to cerebral processes (see [PE02]). However, the contemporary neurobiologist and philosopher Gerhard Roth continues to defend a form of 'non–reductive materialism' [Rot01].

Analytical Psychology

Recall that *analytical psychology* (AP) is part of the *Jungian psychology movement* started by Carl G. Jung[103] and his followers. Although considered to

Sartre calls 'reflective consciousness'. There is no way for the reflective consciousness to subsume the pre–reflective, and so reflection is fated to a form of anxiety, i.e., the human condition. The reflective consciousness in all its forms, (scientific, artistic or otherwise) can only limit the thing–in–itself by virtue of its attempt to understand or describe it. It follows therefore that any attempt at self–knowledge (self–consciousness) is a construct that fails no matter how often it is attempted. (self-consciousness is a reflective consciousness of an overflowing infinite) In Sartre's words "Conciousness is consciousness of itself insofar as it is consciousness of a transcendent object." The same holds true about knowledge of the 'Other' (being), which is a construct of reflective consciousness. One must be careful to understand this more as a form of warning than as an ontological statement. However, there is an implication of Solipsism here that Sartre considers fundamental to any coherent description of the human condition.

[102] Emil du Bois–Reymond (November 7, 1818, Berlin, Germany – November 26, 1896), was a German physician and physiologist, discoverer of the nerve action potential and the father of experimental electrophysiology.

[103] Carl Gustav Jung (July 26, 1875 – June 6, 1961) was a Swiss psychiatrist and founder of *analytical psychology*.

Jung's unique and broadly influential approach to psychology emphasized understanding the *psyche* through exploring the worlds of dreams, art, mythology, world religion and philosophy. Though not the first to analyze dreams, he has become perhaps the best–known pioneer in the field of *dream analysis*. Although he was a theoretical psychologist and practicing clinician for most of his life, much of his life's work was spent exploring other realms: Eastern vs. Western philosophy, alchemy, astrology, sociology, as well as literature and the arts. Jung also emphasized the importance of balance. He cautioned that modern humans rely too heavily on science and logic and would benefit from integrating spirituality and appreciation of the unconscious realm. Interestingly, Jungian ideas are not typically included in curriculum of most major universities' psychology departments, but are occasionally explored in humanities departments. Many pioneering psychological concepts were originally proposed by Jung. Some of these are: (i) *archetype*, (ii) *collective unconscious*, (iii) *unconscious complex*, and (iv) *synchronicity*. In addition, the popular career test currently offered by high school and college career centers, the *Myers–Briggs Type Indicator*, is strongly influenced by Jung's theories.

The overarching goal of Jung's work was the reconciliation of the life of the individual with the world of the *supra–personal archetypes*. He came to see the individual's encounter with the unconscious as central to this process. The human experiences the unconscious through symbols encountered in all aspects of life: in dreams, art, religion, and the symbolic dramas we enact in our relationships and life pursuits. Essential to the encounter with the unconscious, and the reconciliation of the individual's consciousness with this broader world, is learning this symbolic language. Only through attention and openness to this world (which is quite foreign to the modern Western mind) are individuals able to harmonize their lives with these supra–personal archetypal forces. In order to undergo the individuation process, the individual must be open to the parts of oneself beyond one's own ego. In order to do this, the modern individual must pay attention to dreams, explore the world of religion and spirituality, and question the assumptions of the operant societal world–view (rather than just blindly living life in accordance with dominant norms and assumptions).

The collective unconscious could be thought of as the DNA of the human psyche. Just as all humans share a common physical heritage and predisposition towards specific physical forms (like having two legs, a heart, etc.) so do all humans have a common psychological predisposition. However, unlike the quantifiable information that composes DNA (in the form of coded sequences of nucleotides), the collective unconscious is composed of archetypes. In contrast to the objective material world, the subjective realm of archetypes can not be fully plumbed through quantitative modes of research. Instead it can be revealed more fully through an examination of the symbolic communications of the human psyche — in art, dreams, religion, myth, and the themes of human relational/behavioral patterns. Devoting his life to the task of exploring and understanding the collective unconscious, Jung theorized that certain symbolic themes exist across all cultures, all epochs, and in every individual.

The *shadow* is an *unconscious complex* that is defined as the diametrical opposite of the conscious self, the ego. The shadow represents unknown attributes and qualities of the ego. There are constructive and destructive types of shadow. On the destructive side, it often represents everything that the conscious person does not wish to acknowledge within themselves. For instance, someone who identifies as being kind has a shadow that is harsh or unkind. Conversely, an individual who is brutal has a kind shadow. The shadow of persons who are convinced that they are ugly appears to be beautiful. On the constructive side, the shadow may represent hidden positive influences. Jung points to the story of Moses and Al–Khidr in the 18th Book of the Koran as an example. Jung emphasized the importance of being aware of shadow material and incorporating it into conscious awareness, lest one project these attributes on others. The shadow in dreams is often represented by dark figures of the same gender as the dreamer. According to Jung the human being deals with the reality of the shadow in four ways: denial, projection, integration and/or transmutation.

Jung identified the *anima* as being the unconscious feminine component of men and the *animus* as the unconscious masculine component in women. However, this is rarely taken as a literal definition: many modern–day Jungian practitioners believe that every person has both an anima and an animus. Jung stated that the anima and animus act as guides to the unconscious unified *Self*, and that forming an awareness and a connection with the anima or animus is one of the most difficult and rewarding steps in psychological growth. Jung reported that he identified his anima as she spoke to him, as an inner voice, unexpectedly one day. Oftentimes, when people ignore the anima or animus complexes, the anima or animus vies for attention by projecting itself on others. This explains, according to Jung, why we are sometimes immediately attracted to certain strangers: we see our anima or animus in them. Love at first sight is an example of anima and animus projection. Moreover, people who strongly identify with their gender role (e.g., a man who acts aggressively and never cries) have not actively recognized or engaged their anima or animus. Jung attributes human rational thought to be the male nature, while the irrational aspect is considered to be natural female. Consequently, irrationality is the male anima shadow and rationality is the female animus shadow.

There are four primary modes of experiencing the world in Jung's *extrovert/introvert model*: two rational functions: *thinking* and *feeling*, and two perceptive functions: *sensation* and *intuition*. Sensation is the perception of facts. Intuition is the perception of the unseen. Thinking is analytical, deductive cognition. Feeling is synthetic, all–inclusive cognition. In any person, the degree of introversion/extroversion of one function can be quite different to that of another function. Broadly speaking, we tend to work from our most developed function, while we need to widen our personality by developing the others. Related to this, Jung noted that the unconscious often tends to reveal itself most easily through a person's least developed function. The encounter with the unconscious and development of the underdeveloped function(s) thus tend to progress together.

Jung had a profesional relationship with the Nobel lauret physicist Wolfgang Pauli. Their work has been published in the books [PJ55, PJ01] as well as in Jung's famous [Jun80].

be a part of *psychoanalysis*, it is distinct from *Freudian psychoanalysis*.[104] While Freudian psychoanalysis assumes that the repressed material hidden in the unconscious is given by repressed sexual instincts, analytical psychology has a more general approach. There is no preconceived assumption about the unconscious material. The unconscious, for Jungian analysts, may contain repressed sexual drives, but also aspirations, fears, etc.

The aim of AP is the personal experience of the deep forces and motivations underlying human behavior. It is related to the so–called *depth psychology* and *archetypal psychology*. Its basic assumption is that the personal unconscious is a potent part, probably the more active part, of the normal human psyche. Reliable communication between the conscious and unconscious parts of the psyche is necessary for wholeness. Also crucial is the belief that *dreams* show ideas, beliefs, and feelings of which individuals may not be readily aware, but need to be, and that such material is expressed in a personalized vocabulary of visual metaphors. Things 'known but unknown' are contained in the unconscious, and dreams are one of the main vehicles for the unconscious to express them.

AP distinguishes between a *personal and* a *collective unconscious*. The collective unconscious contains *archetypes* common to all human beings. That is, individuation may bring to surface symbols that do not relate to the life experiences of a single person. This content is more easily viewed as answers to the more fundamental questions of humanity: life, death, meaning, happiness, fear. Among these more spiritual concepts may arise and be integrated into the personality.

AP distinguishes two main psychological types or temperaments: (i) *extrovert*, and (ii) *introvert*.[105]The attitude type could be thought of as the *energy*

[104] For a period of some 6 years, Carl Jung was a close friend and collaborator of Sigmund Freud. However after Jung published his 'Wandlungen und Symbole der Libido' (The Psychology of the Unconscious) in 1913, their theoretical ideas had diverged sharply.

[105] In the context of *personality psychology*, *extroverts and introverts* differ in how they get or lose energy as a function of their immediate social context. In particular, extroverts feel an increase of perceived energy when interacting with large group of people, but a decrease of energy when left alone. Conversely, introverts feel an increase of energy when alone, but a decrease of energy when surrounded by large group of people.

Extroverts tend to be energetic when surrounded by people and depressive when not. To induce human interactions, extroverts tend to be enthusiastic, talkative, and assertive. Extroverts enjoy doing activities that involve other people, such as taking part in community activities and involving in business, religious, political, and scientific affairs; their affinity to large groups allow them to enjoy large social gatherings including parties and marches. As such, an extroverted person is likely to enjoy time spent with people and find less reward in time spent alone.

On the other hand, introverts are 'geared to inspect' rather than to act in social settings. In a large social setting, introverts tend to be quiet, low–key,

flow of *libido*, or *psychic energy* (*ch'i* in Roman–Chinese and 'ki' in Roman–Japanese).[106] The introvert's energy flow is inward to the subject and away

deliberate, and engaged in non–social activities. Conversely, introverts gain energy when alone performing solitary activities. Thus they tend to enjoy reading, writing, watching movies at home, inventing, and designing - and doing these activities in quiet, minimally socially interactive environment such as home, library, labs, and quiet coffee shops. While introverts avoid social situations with large numbers of people, they tend to enjoy intense, one–to–one or one–to–few social interactions. They tend to have small circle of very close friends, compared to the extroverts' typically larger circle of less–close friends.

While most people view being either introverted or extroverted as a question with only two answers, levels of extraversion in fact fall in a normally distributed bell curve, with most people falling in between. The term *ambivert* was coined to denote people who fall more or less directly in the middle and exhibit tendencies of both groups. An ambivert is normally comfortable with groups and enjoys social interaction, but also relishes time alone and away from the crowd.

[106] Freud introduced the term *libido* as the instinctual energy or force that can come into conflict with the conventions of civilized behavior. It is the need to conform to society and control the libido, contained in what Freud defined as the Id, that leads to tension and disturbance in both society and the individual. This disturbance Freud labelled neurosis. Thus, libido has to be transformed into socially useful energy, according to Freud, through the process of 'sublimation'.

Ch'i (or *qi,* or *ki*) is a fundamental concept of traditional Chinese culture. Ch'i is believed to be part of everything that exists, as in 'life force' or 'life energy', something like the 'force' in Lucas' Star Wars. It is most often translated as 'energy flow,' or literally as 'air' or 'breath'.

The nature of ch'i is a matter of controversy among those who accept it as a valid concept, while those who dismiss its very existence ignore it, except for purposes of discussion with its adherents. Disputing the nature of qi is an old controversy in Chinese philosophy. Among some traditional Chinese medicine practitioners, qi is sometimes thought of as a metaphor for biological processes similar to the Western concept of energy flow for *homeostatic balance* in biological regulations. Others argue that qi involves some new physics or biology. Attempts to directly connect qi with some scientific phenonomena have been attempted since the mid–nineteenth century. *Ch'i* is a central concept in many martial arts; e.g., in the Japanese arts, Ki is developed in Aikido and given special emphasis in Ki–Aikido (a classic combat story concerns two opponents who held each others hands before a fight, while doing so each felt the others *ch'i* and the one with the weaker *ch'i* resigned without a blow being struck).

The concept of *quantum tunneling* in modern physics where physical matters can 'tunnel' through energy barriers using quantum mechanics captured some of the similar concepts of *ch'i* (which allows one to transcend normal physical forces in nature). The seemingly impossibility of tunneling through energy barriers (walls) is only limited by the conceptual framework of classical mechanics, but can easily be resolved by the *wave–particle duality* in modern physics. By the same token, this duality is similar to the metaphorical duality of *yin* and *yang*, which is governed by the flow of energy *ch'i*. Examples of quantum tunneling can be found as a mechanism in biology used by enzymes to speed up reactions

from the object, i.e., external relations. The extrovert's energy flow is outward toward the object, ie. towards external relations and away from the inner, subjective world. Extroverts desire breadth, while introverts seek depth. The introversion/extroversion attitude type may also influence mental breakdown. Introverts may be more inclined to catatonic type schizophrenia and extroverts towards manic depression.

Samuels [Sam95] has distinguished three schools of 'post–Jungian' psycho–therapy: the classical, the developmental and the archetypal. The classical school is that which tries to remain faithful to what Jung himself proposed and taught in person and in his 20–plus volumes of work. The developmental school, associated with M. Fordham, B. Feldman etc., can be considered a bridge between Jungian psychoanalysis and M. Klein's *object relations theory*. The archetypal school (sometimes called 'the imaginal school'), with different views associated with the *mythopoeticists*, such as J. Hillman in his intellectual theoretical view of archetypal psychology, C.P. Estés, in her view that ethnic and Aboriginal people are the originators of archetypal psychology and have long carried the maps to the journey of the soul in their songs, tales, dream–telling, art and rituals; M. Woodman who proposes a feminist viewpoint regarding archetypal psychology, and other Jungians like T. Moore and R. Moore, as well. Most mythopoeticists/archetypal psychology innovators either imagine the *Self* not to be the main archetype of the collective unconscious as Jung thought, but rather assign each archetype equal value ... Others, who are modern progenitors of archetypal psychology (such as Estés), think of the *Self* as that which contains and yet is suffused by all the other archetypes, each giving life to the other.

1.2 Artificial and Computational Intelligence

1.2.1 Artificial Intelligence

Recall that *artificial intelligence* (AI) is a branch of computer science that deals with *intelligent behavior*, *learning* and *adaptation* in machines. Research in AI is concerned with producing machines to automate tasks requiring intelligent behavior. Examples include control, planning and scheduling, the ability to answer diagnostic and consumer questions, handwriting, speech, and facial recognition. As such, it has become an engineering discipline, focused on providing solutions to real life problems. AI systems are now in routine use in economics, medicine, engineering and the military, as well as being built

in lifeforms to millions of times their normal speed [MRJ06]. Other examples of quantum tunneling are found in semiconductor and superconductors, such as field emission used in flash memory and major source of current leakage in *very–large–scale integration* (VLSI) electronics draining power in mobile phones and computers.

into many common home computer software applications, traditional strategy games like computer chess and other video games.

In the philosophy of artificial intelligence, the so–called *strong AI* is the supposition that some forms of artificial intelligence can truly reason and solve problems; strong AI supposes that it is possible for machines to become sapient, or self–aware, but may or may not exhibit human–like thought processes. The term strong AI was originally coined by John Searle [Sea80]: "According to strong AI, the computer is not merely a tool in the study of the mind; rather, the appropriately programmed computer really is a mind." The term 'artificial intelligence' would equate to the same concept as what we call 'strong AI' based on the literal meanings of 'artificial' and 'intelligence'. However, initial research into artificial intelligence was focused on narrow fields such as pattern recognition and automated scheduling, in hopes that they would eventually allow for an understanding of true intelligence. The term 'artificial intelligence' thus came to encompass these narrower fields, the so–called *weak AI* as well as the idea of strong AI.

In contrast to strong AI, weak AI refers to the use of software to study or accomplish specific problem solving or reasoning tasks that do not encompass (or in some cases, are completely outside of) the full range of human cognitive abilities. An example of weak AI software would be a chess program such as *Deep Blue*. Unlike strong AI, a weak AI does not achieve self–awareness or demonstrate a wide range of human–level cognitive abilities, and at its finest is merely an intelligent, more specific problem–solver. Some argue that weak AI programs cannot be called 'intelligent' because they cannot really think.

AI divides roughly into two schools of thought: *Conventional AI* and *Computational Intelligence* (CI). Conventional AI mostly involves methods now classified as machine learning, characterized by formalism and statistical analysis. This is also known as symbolic AI, logical AI, neat AI and good old–fashioned AI (which mainly deals with symbolic problems). Basic Ai methods include:

1. Expert systems: apply reasoning capabilities to reach a conclusion. An expert system can process large amounts of known information and provide conclusions based on them.
2. Case based reasoning
3. Bayesian networks
4. Behavior based AI: a modular method of building AI systems by hand.

On the other hand, CI involves iterative development or learning (e.g., parameter tuning in connectionist systems). Learning is based on empirical data and is associated with non–symbolic AI and soft computing. Methods mainly include:

1. Neural networks: systems with very strong pattern recognition capabilities;

2. Fuzzy systems: techniques for reasoning under uncertainty, have been widely used in modern industrial and consumer product control systems; and
3. Evolutionary computation: applies biologically inspired concepts such as populations, mutation and survival of the fittest to generate increasingly better solutions to the problem. These methods most notably divide into evolutionary algorithms (e.g. genetic algorithms) and swarm intelligence (e.g. ant algorithms).

With hybrid intelligent systems attempts are made to combine these two groups. Expert inference rules can be generated through neural network or production rules from statistical learning such as in ACT–R.

A promising new approach called intelligence amplification tries to achieve artificial intelligence in an evolutionary development process as a side–effect of amplifying human intelligence through technology.

Brief AI History

Early in the 18th century, René Descartes envisioned the bodies of animals as complex but reducible machines, thus formulating the mechanistic theory, also known as the 'clockwork paradigm'. Wilhelm Schickard created the first mechanical digital calculating machine in 1623, followed by machines of Blaise Pascal[107] (1643) and Gottfried Wilhelm von Leibniz (1671), who also invented the binary system. In the 19th century, Charles Babbage and Ada Lovelace worked on programmable mechanical calculating machines.

Bertrand Russell and Alfred Whitehead published their 'Principia Mathematica' in 1910–1913, which revolutionized *formal logic*. In 1931 Kurt Gödel showed that sufficiently powerful consistent formal systems contain true theorems unprovable by any theorem–proving AI that is systematically deriving all possible theorems from the axioms. In 1941 Konrad Zuse built the first working program–controlled computers. Warren McCulloch and Walter Pitts published A Logical Calculus of the Ideas Immanent in Nervous Activity

[107] Blaise Pascal (June 19, 1623 – August 19, 1662) was a French mathematician, physicist, and religious philosopher. Pascal was a child prodigy, who was educated by his father. Pascal's earliest work was in the natural and applied sciences, where he made important contributions to the construction of mechanical calculators and the study of fluids, and clarified the concepts of pressure and vacuum by expanding the work of Evangelista Torricelli. Pascal also wrote powerfully in defense of the scientific method.

He was a mathematician of the first order. Pascal helped create two major new areas of research. He wrote a significant treatise on the subject of projective geometry at the age of sixteen and corresponded with Pierre de Fermat from 1654 on probability theory, strongly influencing the development of modern economics and social science.

(1943), laying the foundations for neural networks. Norbert Wiener's 'Cybernetics or Control and Communication in the Animal and the Machine' (MIT Press, 1948) popularizes the term 'cybernetics'.

The 1950s were a period of active efforts in AI. In 1950, Alan Turing introduced the 'Turing test' as a way of operationalizing a test of intelligent behavior. The first working AI programs were written in 1951 to run on the Ferranti Mark I machine of the University of Manchester: a draughts–playing program written by Christopher Strachey and a chess–playing program written by Dietrich Prinz. John McCarthy coined the term 'artificial intelligence' at the first conference devoted to the subject, in 1956. He also invented the Lisp programming language. Joseph Weizenbaum built ELIZA, a chatterbot implementing Rogerian psychotherapy. At the same time, John von Neumann,[108] who had been hired by the RAND Corporation, developed the *game theory*, which would prove invaluable in the progress of AI research.

During the 1960s and 1970s, Joel Moses demonstrated the power of symbolic reasoning for integration problems in the Macsyma program, the first successful knowledge–based program in mathematics. Leonard Uhr and Charles Vossler published 'A Pattern Recognition Program That Generates, Evaluates, and Adjusts Its Own Operators' in 1963, which described one of the first machine learning programs that could adaptively acquire and modify features and thereby overcome the limitations of simple perceptrons of Frank Rosenblatt.[109] Marvin Minsky and Seymour Papert published their

[108] John von Neumann (Neumann János) (December 28, 1903 – February 8, 1957) was an Austro–Hungarian mathematician and polymath who made contributions to quantum physics, functional analysis, set theory, game theory, economics, computer science, topology, numerical analysis, hydrodynamics (of explosions), statistics and many other mathematical fields as one of world history's outstanding mathematicians. His PhD supervisor was David Hilbert. Most notably, von Neumann was a pioneer of the modern digital computer and the application of operator theory to quantum mechanics, a member of the Manhattan Project and the first faculty of the Institute for Advanced Study at Princeton (along with Albert Einstein and Kurt Gödel), and creator of *game theory* and the concept of *cellular automata*. Along with Edward Teller and Stanislaw Ulam, von Neumann worked out key steps in the nuclear physics involved in thermonuclear reactions and the hydrogen bomb.

[109] Frank Rosenblatt (1928–1969) was a New York City born computer scientist who completed the Perceptron (the simplest kind of feedforward neural network: a linear classifier) on MARK 1, computer at Cornell University in 1960. This was the first computer that could learn new skills by trial and error, using a type of neural network that simulates human thought processes.

Rosenblatt's perceptrons were initially simulated on an IBM 704 computer at Cornell Aeronautical Laboratory in 1957. By the study of neural networks such as the Perceptron, Rosenblatt hoped that "the fundamental laws of organization which are common to all information handling systems, machines and men included, may eventually be understood."

book 'Perceptrons', which demonstrated the limits of simple neural nets. Alain Colmerauer developed the Prolog computer language. Ted Shortliffe demonstrated the power of rule–based systems for knowledge representation and inference in medical diagnosis and therapy in what is sometimes called the first expert system. Hans Moravec developed the first computer–controlled vehicle to autonomously negotiate cluttered obstacle courses.

In the 1980s, neural networks became widely used due to the *backpropagation algorithm*, first described by Paul Werbos in 1974. The team of Ernst Dickmanns built the first robot cars, driving up to 55 mph on empty streets. The 1990s marked major achievements in many areas of AI and demonstrations of various applications. In 1995, one of Dickmanns' robot cars drove more than 1000 miles in traffic at up to 110 mph. Deep Blue, a chess–playing computer, beat Garry Kasparov in a famous six–game match in 1997. DARPA stated that the costs saved by implementing AI methods for scheduling units in the first Persian Gulf War have repaid the US government's entire investment in AI research since the 1950s. Honda built the first prototypes of humanoid robots.

During the 1990s and 2000s AI has become very influenced by probability theory and statistics. Bayesian networks are the focus of this movement, providing links to more rigorous topics in statistics and engineering such as Markov models and Kalman filters, and bridging the old divide between 'neat' and 'scruffy' approaches. The last few years have also seen a big interest in game theory applied to AI decision making. This new school of AI is sometimes called 'machine learning'. After the September 11, 2001 attacks there has been much renewed interest and funding for threat–detection AI systems, including machine vision research and data–mining. The DARPA Grand Challenge is a race for a $2 million prize where cars drive themselves across several hundred miles of challenging desert terrain without any communication with humans, using GPS, computers and a sophisticated array of sensors. In 2005 the winning vehicles completed all 132 miles of the course.

Cybernetics, General Systems Theory and Bionics

Closely related to AI is *cybernetics*, which is the study of communication and control, typically involving regulatory feedback, in living organisms, in machines and organisations and their combinations, for example, in sociotechnical systems, computer controlled machines such as automata and robots. The term *cybernetics* stems from the Greek 'kybernetes', which means steersman, governor, pilot, or rudder, which has the same root as government. It is an earlier but still–used generic term for many of the subject matters that are increasingly subject to specialization under the headings of adaptive systems, artificial intelligence, complex systems, complexity theory, control systems, decision support systems, dynamical systems, information theory, learning organizations, mathematical systems theory, operations research, simulation, and systems engineering.

Contemporary cybernetics began as an interdisciplinary study connecting the fields of control systems, electrical network theory, logic modeling, and neuroscience in the 1940s. The name cybernetics was coined by Norbert Wiener[110] to denote the study of 'teleological mechanisms' and was popularized through his book 'Cybernetics, or Control and Communication in the Animal and Machine' (MIT, 1948).

The study of *teleological mechanisms* in machines with *corrective feedback* dates from as far back as the late 1700s when James Watt's steam engine was equipped with a governor, a centrifugal feedback valve for controlling the speed of the engine. In 1868 James Clerk Maxwell[111] published a theoretical article on governors. In 1935 Russian physiologist P.K. Anokhin published a book 'Physiology of Functional Systems' on in which the concept of feedback ('back afferentation') was studied. In the 1940s the study and mathematical modelling of regulatory processes became a continuing research effort and two key articles were published in 1943. These papers were 'Behavior, Purpose and Teleology' by Rosenblueth, Wiener and Bigelow; and the paper 'A Logical Calculus of the Ideas Immanent in Nervous Activity' by McCulloch and Pitts.

[110] Norbert Wiener (November 26, 1894 – March 18, 1964) was an American mathematician and applied mathematician, especially in the field of electronics engineering. He was a pioneer in the study of *stochastic processes* (random processes) and noise processes, especially in the field of *electronic communication systems* and *control systems*. He is known as the founder of cybernetics. He coined the term 'cybernetics' in his book 'Cybernetics or Control and Communication in the Animal and the Machine' (MIT Press, 1948), widely recognized as one of the most important books of contemporary scientific thinking. He is also considered by some to be the first American–born–and–trained mathematician on an intellectual par with the traditional bastions of mathematical learning in Europe. He thus represents a watershed period in American mathematics. Wiener did much valuable work in defense systems for the United States, particularly during World War II and the Cold War.

[111] James Clerk Maxwell (13 June 1831 – 5 November 1879) was a Scottish mathematical physicist, born in Edinburgh. Maxwell formulated a set of equations expressing the basic laws of *electricity and magnetism* and developed the Maxwell distribution in the *kinetic theory of gases*. He is also credited with developing the first permanent colour photograph in 1861.

Maxwell had one of the finest mathematical minds of any theoretical physicist of his time. Maxwell is widely regarded as the nineteenth century scientist who had the greatest influence on twentieth century physics, making contributions to the fundamental models of nature. In 1931, on the centennial anniversary of Maxwell's birthday, Einstein described Maxwell's work as the "most profound and the most fruitful that physics has experienced since the time of Newton."

Algebraic mathematics with elements of geometry are a feature of much of Maxwell's work. Maxwell demonstrated that electric and magnetic forces are two complementary aspects of electromagnetism. He showed that electric and magnetic fields travel through space, in the form of waves, at a constant velocity of 3.0×108 m/s. He also proposed that light was a form of electromagnetic radiation.

Wiener himself popularized the social implications of cybernetics, drawing analogies between automatic systems such as a regulated steam engine and human institutions in his best–selling 'The Human Use of Human Beings: Cybernetics and Society' (Houghton–Mifflin, 1950).

In scholarly terms, cybernetics is the study of systems and control in an abstracted sense, that is, it is not grounded in any one empirical field. The emphasis is on the functional relations that hold between the different parts of a system, rather than the parts themselves. These relations include the transfer of *information*, and circular relations (*feedbacks*) that result in emergent phenomena such as *self–organization*. The main innovation of cybernetics was the creation of a scientific discipline focused on goals: an understanding of goal–directedness or purpose, resulting from a *negative feedback loop* which minimizes the deviation between the perceived situation and the desired situation (goal). As mechanistic as that sounds, cybernetics has the scope and rigor to encompass the human social interactions of agreement and collaboration that, after all, require goals and feedback to attain (see, e.g., [Ash56]). Related to cybernetics are: engineering cybernetics, quantum cybernetics, biological cybernetics, medical cybernetics, psychocybernetics, sociocybernetics and organizational cybernetics.

On the other hand, *general systems theory* is an interdisciplinary field that studies the properties of systems as a whole. It was founded by Ludwig von Bertalanffy, Ross W. Ashby, Margaret Mead, Gregory Bateson and others in the 1950s. Also, John von Neumann discovered cellular automata and self–reproducing systems without computers, with only pencil and paper. Aleksandr Lyapunov and Jules Henri Poincaré worked on the foundations of chaos theory without any computer at all. Ilya Prigogine, Prigogine has studied 'far from equilbrium systems' for emergent properties, suggesting that they offer analogues for living systems.

Systems theory brought together theoretical concepts and principles from ontology, philosophy of science, physics, biology and engineering and later found applications in numerous fields including geography, sociology, political science, organizational theory, management, psychotherapy (within family systems therapy) and economics among others. Cybernetics is a closely related field. In recent times systems science, systemics and complex systems have been used as synonyms.

Cybernetics, catastrophe theory and chaos theory have the common goal to explain complex systems that consist of a large number of mutually interacting and interrelated parts in terms of those interactions. Cellular automata (CA), neural networks (NN), artificial intelligence (AI), and artificial life (ALife) are related fields, but they do not try to describe general(universal) complex (singular) systems. The best context to compare the different "C"-Theories about complex systems is historical, which emphasizes different tools and methodologies, from pure mathematics in the beginning to pure computer science now. Since the beginning of chaos theory when Edward Lorenz accidentally

discovered a *strange attractor*[112] with his computer, computers have become an indispensable source of information. One could not imagine the study of complex systems without computers today.

In recent years, the field of systems thinking has been developed to provide techniques for studying systems in holistic ways to supplement more traditional reductionistic methods. In this more recent tradition, systems theory is considered by some as a humanistic extension of the natural sciences.

Finally, bionics is the application of methods and systems found in nature to the study and design of engineering systems and modern technology. Also a short form of biomechanics, the word 'bionic' is actually a portmanteau formed from biology and electronic.

The transfer of technology between lifeforms and synthetic constructs is desirable because evolutionary pressure typically forces natural systems to become highly optimized and efficient. A classical example is the development of dirt– and water–repellent paint (coating) from the observation that the surface of the lotus flower plant is practically unsticky for anything (the lotus effect). Examples of bionics in engineering include the hulls of boats imitating the thick skin of dolphins, sonar, radar, and medical ultrasound imaging imitating the echolocation of bats.

In the field of computer science, the study of bionics has produced cybernetics, artificial neurons, artificial neural networks, and swarm intelligence. Evolutionary computation was also motivated by bionics ideas but it took the idea further by simulating evolution in silico and producing well-optimized solutions that had never appeared in nature.

Often, the study of bionics emphasizes imitation of a biological structure rather than just an implementation of its function. The conscious copying of examples and mechanisms from natural organisms and ecologies is a form of applied case–based reasoning, treating nature itself as a database of solutions that already work. Proponents argue that the selective pressure placed on all natural life forms minimizes and removes failures.

Roughly, we can distinguish three biological levels in biology after which technology can be modelled:

1. mimicking natural methods of manufacture of chemical compounds to create new ones;
2. imitating mechanisms found in nature; and
3. studying organizational principles from social behaviour of organisms, such as the flocking behaviour of birds or the emergent behaviour of bees and ants.

[112] *Strange attractor* is an attracting set that has zero measure in the embedding phase–space and has fractal dimension. Trajectories within a strange attractor appear to skip around randomly (see Chapter 2 for details).

Turing Test and General AI

Recall that the *Turing test* is a proposal for a test of a machine's capability to perform human–like conversation. Described by Alan Turing[113] in his 1950 paper 'Computing machinery and intelligence,'[114] it proceeds as follows: a human judge engages in a natural language conversation with two other parties, one a human and the other a machine; if the judge cannot reliably tell which is which, then the machine is said to pass the test. It is assumed that both the human and the machine try to appear human. In order to keep the test setting simple and universal (to explicitly test the linguistic capability of the machine instead of its ability to render words into audio), the conversation is usually limited to a text–only channel such as a teletype machine as Turing suggested or, more recently IRC or instant messaging.

General artificial intelligence research aims to create AI that can *replicate human intelligence completely*, often called an Artificial General Intelligence (AGI) to distinguish from less ambitious AI projects. As yet, researchers have devoted little attention to AGI, many claiming intelligence is too complex to be completely replicated. Some small groups of computer scientists are doing some AGI research, however. By most measures, demonstrated progress towards strong AI has been limited, as no system can pass a full Turing test for unlimited amounts of time, although some AI systems can at least fool some people initially now (see the Loebner prize winners). Few active AI researchers are prepared to publicly predict whether, or when, such systems will be developed, perhaps due to the failure of bold, unfulfilled predictions for AI research progress in past years. There is also the problem of the AI

[113] Alan Mathison Turing, OBE (June 23, 1912 – June 7, 1954) was an English mathematician, logician, and cryptographer. Turing is often considered to be the father of modern computer science.

With the Turing test, Turing made a significant and characteristically provocative contribution to the debate regarding artificial intelligence: whether it will ever be possible to say that a machine is conscious and can think. He provided an influential formalisation of the concept of algorithm and computation with the Turing machine, formulating the now widely accepted "Turing" version of the Church–Turing thesis, namely that any practical computing model has either the equivalent or a subset of the capabilities of a Turing machine. During World War II, Turing worked at Bletchley Park, Britain's codebreaking centre and was for a time head of Hut 8, the section responsible for German Naval cryptanalysis. He devised a number of techniques for breaking German ciphers, including the method of the bombe, an electromechanical machine which could find settings for the Enigma machine.

[114] In Turing's paper, the term 'Imitation Game' is used for his proposed test as well as the party game for men and women. The name 'Turing test' may have been invented, and was certainly publicized, by Arthur C. Clarke in the science–fiction novel 2001: A Space Odyssey (1968), where it is applied to the computer HAL 9000.

effect, where any achievement by a machine tends to be deprecated as a sign of true intelligence.

Computer Simulation of Human Brain

This is seen by many as the quickest means of achieving Strong AI, as it doesn't require complete understanding. It would require three things:

1. Hardware: an extremely powerful computer would be required for such a model. Futurist Ray Kurzweil estimates 1 million MIPS. If *Moore's law* continues, this will be available for £1000 by 2020.
2. Software: this is usually considered the hard part. It assumes that the human mind is the central nervous system and is governed by physical laws.
3. Understanding: finally, it requires sufficient understanding thereof to be able to model it mathematically. This could be done either by understanding the central nervous system, or by mapping and copying it. Neuro–imaging technologies are improving rapidly, and Kurzweil predicts that a map of sufficient quality will become available on a similar timescale to the required computing power.

Once such a model is built, it will be easily altered and thus open to trial and error experimentation. This is likely to lead to huge advances in understanding, allowing the model's intelligence to be improved/motivations altered. Current research in the area is using one of the fastest supercomputer architectures in the world, namely the Blue Gene platform created by IBM to simulate a single Neocortical Column consisting of approximately 60,000 neurons and 5km of interconnecting synapses. The eventual goal of the project is to use supercomputers to simulate an entire brain.

In opposition to human–brain simulation, the direct approach attempts to achieve AI directly without imitating nature. By comparison, early attempts to construct flying machines modelled them after birds, but modern aircraft do not look like birds. The main question in the direct approach is: 'What is AI?'. The most famous definition of AI was the operational one proposed by Alan Turing in his 'Turing test' proposal (see footnote above). There have been very few attempts to create such definition since (some of them are in the AI Project). John McCarthy[115] stated in his work 'What is AI?' that we

[115] John McCarthy (born September 4, 1927, in Boston, Massachusetts, sometimes known affectionately as Uncle John McCarthy), is a prominent computer scientist who received the Turing Award in 1971 for his major contributions to the field of Artificial Intelligence. In fact, he was responsible for the coining of the term 'Artificial Intelligence' in his 1955 proposal for 1956 Dartmouth Conference.

McCarthy championed expressing knowledge declaratively in mathematical logic for Artificial Intelligence. An alternative school of thought emerged at MIT and elsewhere proposing the 'procedural embedding of knowledge' using high

still do not have a solid definition of intelligence (compare with the previous section).

Machine Learning

As a broad AI–subfield, *machine learning* (ML) is concerned with the development of algorithms and techniques that allow computers to 'learn'. At a general level, there are two types of learning: inductive, and deductive. Inductive machine learning methods create computer programs by extracting rules and patterns out of massive data sets. It should be noted that although pattern identification is important to ML, without rule extraction a process falls more accurately in the field of *data mining*.

Machine learning overlaps heavily with statistics. In fact, many machine learning algorithms have been found to have direct counterparts with statistics. For example, boosting is now widely thought to be a form of stagewise regression using a specific type of loss function.

Machine learning has a wide spectrum of applications including search engines, medical diagnosis, bioinformatics and cheminformatics, detecting credit card fraud, stock market analysis, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing and robot locomotion.

Some machine learning systems attempt to eliminate the need for human intuition in the analysis of the data, while others adopt a collaborative approach between human and machine. Human intuition cannot be entirely eliminated since the designer of the system must specify how the data are to be represented and what mechanisms will be used to search for a characterization of the data. Machine learning can be viewed as an attempt to automate parts of the scientific method. Some machine learning researchers create methods within the framework of *Bayesian statistics*.

---

level plans, assertions, and goals first in Planner and later in the Scientific Community Metaphor. The resulting controversy is still ongoing and the subject matter of research.

McCarthy invented the Lisp programming language and published its design in Communications of the ACM in 1960. He helped to motivate the creation of Project MAC at MIT, but left MIT for Stanford University in 1962, where he helped set up the Stanford AI Laboratory, for many years a friendly rival to Project MAC.

In 1961, he was the first to publicly suggest (in a speech given to celebrate MIT's centennial) that computer time–sharing technology might lead to a future in which computing power and even specific applications could be sold through the utility business model (like water or electricity). This idea of a computer or information utility was very popular in the late 1960s, but faded by the mid–1970s as it became clear that the hardware, software and telecommunications technologies of the time were simply not ready. However, since 2000, the idea has resurfaced in new forms.

Machine learning algorithms are organized into a taxonomy, based on the desired outcome of the algorithm. Common algorithm types include:

1. *supervised learning*, where the algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behavior of) a function which maps a vector into one of several classes by looking at several input–output examples of the function.
2. *unsupervised/self–organized learning*, which models a set of inputs: labeled examples are not available.
3. *semi–supervised learning*, which combines both labeled and unlabeled examples to generate an appropriate function or classifier.
4. *reinforcement learning*, where the algorithm learns a policy of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm.
5. *transduction*, similar to supervised learning, but does not explicitly construct a function: instead, tries to predict new outputs based on training inputs, training outputs, and new inputs.
6. *learning to learn*, where the algorithm learns its own inductive bias based on previous experience.

Symbol–Based Learning

The *symbol–based learning* relies on learning algorithms that can be characterized into the following five dimensions [Lug02]:

– *data and goals*: here the learning problem is described according to the goals of the learner and the data it is initially given;
– *knowledge representation*: using representation languages with programs to store the knowledge learned by the system in a logical way;
– *learning operations*: an agent is given a set of training instances and it is tasked to construct a generalization, heuristic rule or a plan that satisfies its goals;
– *concept space*: the representation language along with the learning operations define a space of possible concept definitions, the learner needs to search this space to find the desired concept. The complexity of this concept space is used to measure how difficult the problem is; and
– *heuristic search*: heuristics are used to commit to a particular direction when searching the concept space.

Connectionist Learning

The *connectionist learning* is performed using artificial neural networks (see subsection below), which are systems comprised of a large number of interconnected artificial neurons. They have been widely used for (see, e.g., [Hay94, Kos92, Lug02]):

– *classification*: deciding the category or grouping where an input value belongs;
– *pattern recognition*: identifying a structure in sometimes noisy data;
– *memory recall*: addressing the content in memory;
– *prediction/forecasting*: identifying an effect from different causes;
– *optimization*: finding the best organization within different constraints; and
– *noise filtering*: separating a signal from the background noise or removing irrelevant components to a signal.

The knowledge of the network is encapsulated within the organization and interaction of the neurons. Specifically, the global properties of neurons are characterized as:

– *network topology*: the topology of the network is the pattern of connections between neurons;
– *learning algorithm*: the algorithm used to change the weight between different connections; and
– *encoding scheme*: the interpretation of input data presented to the network and output data obtained from the network.

Learning is achieved by modifying the structure of the neural network, via adjusting weights, in order to map input combinations to required outputs. There are two general classes of learning algorithms for training neural networks, they are supervised and unsupervised learning. Supervised learning requires the neural network to have a set of training data, consisting of the set of data to be learned as well as the corresponding answer. The data set is repeatedly presented to the neural network, in turn, the network adapts by changing the weights of connections between the neurons until the network output corresponds closely to the required answers. The goal of supervised learning is to find a model or mapping that will correctly associate its inputs with its targets. Supervised learning is suited to applications when the outputs expected from the network are well known. This allows the designer (or another fully trained network) to provide feedback.

In the case of unsupervised learning the target value is not provided and the information in the training data set is continuously presented until some convergence criteria is satisfied. This involves monitoring the output of the network and stopping its training when some desired output is observed. The main difference to supervised learning is that the desired output is not known when the training starts. During training, the network has to continuously adapt and change its output until it demonstrates a useful output behavior at which time it receives a single feedback to stop. The input data provided to the network will need to include sufficient information so that the problem is unambiguous. Unsupervised learning is suitable in situations where there is no clear–cut answer to a given problem.

The biggest problem of using neural networks with agents with that the concepts cannot intuitively fit within the agent oriented paradigm. However,

neural networks have been used to implement part of a system such as pattern recognition and classification. It is also believed that neural learning concepts and techniques will play an important role in future research [Lug02].

Computational Learning Theory

The performance and computational analysis of machine learning algorithms is a branch of statistics known as *computational learning theory*. Machine learning algorithms take a training set, form hypotheses or models, and make predictions about the future. Because the training set is finite and the future is uncertain, learning theory usually does not yield absolute guarantees of performance of the algorithms. Instead, probabilistic bounds on the performance of machine learning algorithms are quite common. In addition to performance bounds, computational learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results (see, e.g., [Ang92]):

1. positive results, showing that a certain class of functions is learnable in polynomial time.
2. negative results, showing that certain classes cannot be learned in polynomial time.

Negative results are proven only by assumption. The assumptions that are common in negative results are:

(i) *computational complexity*: $P \neq NP$,[116] and

[116] The relationship between the complexity classes P and NP is an unsolved question in theoretical computer science. It is generally agreed to be the most important such unsolved problem, and one of the most important unsolved problems in all of mathematics. The Clay Mathematics Institute has offered a US $1,000,000 prize for a correct solution.

In essence, the $P = NP$ question asks: if positive solutions to a YES/NO problem can be verified quickly, can the answers also be computed quickly? Consider, for instance, the subset–sum problem, an example of a problem which is easy to verify, but is believed (but not proved) to be difficult to compute the answer. Given a set of integers, does any subset of them sum to 0? For instance, does a subset of the set $\{-2, -3, 15, 14, 7, -10\}$ add up to 0? The answer is YES, though it may take a little while to find a subset that does – and if the set was larger, it might take a very long time to find a subset that does. On the other hand, if someone claims that the answer is "YES, because $\{-2, -3, -10, 15\}$ add up to zero," then we can quickly check that with a few additions. Verifying that the subset adds up to zero is much faster than finding the subset in the first place. The information needed to verify a positive answer is also called a certificate. So we conclude that given the right certificates, positive answers to our problem can be verified quickly (i.e. in polynomial time) and that's why this problem is in NP.

(ii) *cryptography*:[117] *one–way functions* exist.

Recall that a one–way function is a function that is easy to calculate but hard to invert, i.e., it is difficult to calculate the input to the function given its output. The precise meanings of 'easy' and 'hard' can be specified mathematically. With rare exceptions, almost the entire field of public key cryptography rests on the existence of one–way functions. Formally, two variants of one–way functions are defined: strong and weak one–way functions:

An answer to the $P = NP$ question would determine whether problems like SUBSET–SUM are really harder to compute than to verify (this would be the case if P does not equal NP), or that they are as easy to compute as to verify (this would be the case if $P = NP$). The answer would apply to all such problems, not just the specific example of SUBSET–SUM.

The restriction to YES/NO problems doesn't really make a difference; even if we allow more complicated answers, the resulting problem (whether $FP = FNP$) is equivalent.

[117] Recall that cryptography (or cryptology; derived from Greek 'kryptós–hidden' and 'gráfein–to write') is a mathematical discipline concerned with information security and related issues, particularly encryption, authentication, and access control. Its purpose is to hide the meaning of a message rather than its existence. In modern times, it has also branched out into computer science. Cryptography is central to the techniques used in computer and network security for such things as *access control* and *information confidentiality*. Cryptography is used in many applications that touch everyday life; the security of ATM cards, computer passwords, and electronic commerce all depend on cryptography.

The so–called *symmetric–key cryptography* refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until 1976.

The modern study of symmetric–key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications (see, e.g., [Gol01]). A block cipher is the modern embodiment of Alberti's polyalphabetic cipher: block ciphers take as input a block of plaintext and a key, and output a block of ciphertext of the same size. Block ciphers are used in a mode of operation to implement a cryptosystem. DES and AES are block ciphers which have been designated cryptography standards by the US government (though DES's designation was eventually withdrawn after the AES was adopted)[8]. Despite its delisting as an official standard, DES (especially its still-approved and much more secure triple–DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to e–mail privacy and secure remote access. Many other block ciphers have been designed and released, with considerable variation in quality. Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit by bit or character by character, somewhat like the one–time pad. In a stream cipher, the output stream is created based on an internal state which changes as the cipher operates. That state's change is controlled by the key, and, in some stream ciphers, by the plaintext stream as well.

1. Strong one–way functions. A function

$$f : \{0,1\}^* \to \{0,1\}^*$$

is called (strongly) one–way if the following two conditions hold: (i) easy to compute: there exists a (deterministic) polynomial–time algorithm A, such that for input x algorithm A outputs $f(x)$ (i.e., $A(x) = f(x)$); and (ii) hard to invert: for any probabilistic polynomial–time algorithm A', and any polynomial $p(\cdot)$, and for sufficiently large n,

$$P(A'(f(U_n), 1^n) \in f^{-1}|f(U_n)) < \frac{1}{p(n)},$$

where U_n denotes a random variable uniformly distributed over $\{0,1\}^n$. Hence, the probability in the second condition is taken over all the possible values assigned to U_n and all possible internal coin tosses of A' with uniform probability distribution. In addition to an input in the range of f the inverting algorithm is also given the desired length of the output in unary notation. The main reason for this convention is to rule out the possibility that a function is considered one–way merely because the inverting algorithm does not have enough time to print the output. The left hand part of the comparison is quite easy to understand: it is the probability, that A' finds any value U, with property $f(U) = f(Un)$. So, basically, the hard–to–invert condition requires this probability to be negligibly small.
2. Weak one–way functions only require that all efficient inverting algorithms fail with some non–negligible probability. A function

$$f : \{0,1\}^* \to \{0,1\}^*$$

is called weakly one–way if the following two conditions hold: (i) easy to compute: as in the definition of strong one–way functionl and (ii) slightly–hard to invert: There exists a polynomial such that for every probabilistic polynomial–time algorithm, A', and all sufficiently large n's,

$$P(A'(f(U_n), 1^n) \notin f^{-1}|f(U_n)) > \frac{1}{p(n)}$$

It is not known whether one–way functions exist. In fact, their existence would imply $P \neq NP$, resolving the foremost unsolved question of computer science. However, it is not clear if $P \neq NP$ implies the existence of one–way functions. It can be proved that weak one–way functions exist if and only if strong one-way functions do. Thus, as far as the mere existence of one–way function goes, the notions of weak and strong one–way functions are equivalent. It is known that the existence of one–way functions implies the existence of many other useful cryptographic primitives, including:

1. Pseudorandom bit generators;
2. Pseudorandom function families;
3. Digital signature schemes (secure against adaptive chosen–message attack).

In particular, a *trapdoor one–way function* (or, trapdoor permutation) is a special kind of one–way function. Such a function is hard to invert unless some secret information, called the trapdoor, is known. RSA is a well known example of a function believed to belong to this class.

Now, there are several different approaches to computational learning theory, which are often mathematically incompatible. This incompatibility arises from using different inference principles: principles which tell us how to generalize from limited data. The incompatibility also arises from differing definitions of probability (see frequency probability, Bayesian probability). The different approaches include:

1. *probably approximately correct learning* (PAC learning),[118] proposed by Leslie Valiant;
2. *statistical learning theory* (or VC theory),[119] proposed by Vladimir Vapnik;

[118] Probably approximately correct learning (PAC learning) is a framework of learning that was proposed by Leslie Valiant in his paper 'A theory of the learnable'. In this framework the learner gets samples that are classified according to a function from a certain class. The aim of the learner is to find a bounded approximation (approximately) of the function with high probability (probably). We demand the learner to be able to learn the concept given any arbitrary approximation ratio, probability of success or distribution of the samples. The model was further extended to treat noise (misclassified samples). The PAC framework allowed accurate mathematical analysis of learning. Also critical are definitions of efficiency. In particular, we are interested in finding efficient classifiers (time and space requirements bounded to a polynomial of the example size) with efficient learning procedures (requiring an example count bounded to a polynomial of the concept size, modified by the approximation and likelihood bounds).

[119] Vapnik–Chervonenkis theory (also known as VC theory, or statistical learning theory) was developed during 1960–1990 by Vladimir Vapnik and Alexey Chervonenkis. The theory is a form of computational learning theory, which attempts to explain the learning process from a statistical point of view. VC theory covers four parts:

a) Theory of consistency of learning processes – what are (necessary and sufficient) conditions for consistency of a learning process based on the empirical risk minimization principle?
b) Nonasymptotic theory of the rate of convergence of learning processes – how fast is the rate of convergence of the learning process?
c) Theory of controlling the generalization ability of learning processes – how can one control the rate of convergence (the generalization ability) of the learning process?
d) Theory of constructing learning machines – how can one construct algorithms that can control the generalization ability?

3. *Bayesian inference* (see below), arising from work first done by Thomas Bayes;[120] and
4. *algorithmic learning theory*,[121] from the work of Mark Gold.

The last part of VC theory introduced a well–known learning algorithm: the *support vector machine*. VC theory contains important concepts such as the VC dimension and structural risk minimization.

[120] Thomas Bayes (c. 1702 – April 17, 1761) was a British mathematician and Presbyterian minister, known for having formulated a special case of Bayes' theorem, which was published posthumously. Bayes' solution to a problem of 'inverse probability' was presented in the Essay Towards Solving a Problem in the Doctrine of Chances (1764), published posthumously by his friend Richard Price in the Philosophical Transactions of the Royal Society of London. This essay contains a statement of a special case of Bayes' theorem.

Bayesian probability is the name given to several related interpretations of probability, which have in common the application of probability to any kind of statement, not just those involving random variables. 'Bayesian' has been used in this sense since about 1950.

It is not at all clear that Bayes himself would have embraced the very broad interpretation now called Bayesian. It is difficult to assess Bayes' philosophical views on probability, as the only direct evidence is his essay, which does not go into questions of interpretation. In the essay, Bayes defines probability as follows:

"The probability of any event is the ratio between the value at which an expectation depending on the happening of the event ought to be computed, and the chance of the thing expected upon it's happening."

In modern *utility theory* we would say that expected utility is the probability of an event times the payoff received in case of that event. Rearranging that to solve for the probability, we get Bayes' definition. As Stigler points out, this is a subjective definition, and does not require repeated events; however, it does require that the event in question be observable, for otherwise it could never be said to have 'happened' (some would argue, however, that things can happen without being observable).

The search engine Google, and the information retrieval company Autonomy Systems, employ Bayesian principles to provide probable results to searches. Microsoft is reported as using Bayesian probability in its future Notification Platform to filter unwanted messages.

In statistics, empirical Bayes methods involve:

a) An 'underlying' probability distribution of some unobservable quantity assigned to each member of a statistical population. This quantity is a random variable if a member of the population is chosen at random. The probability distribution of this random variable is not known, and is thought of as a property of the population.
b) An observable quantity assigned to each member of the population. When a random sample is taken from the population, it is desired first to estimate the "underlying" probability distribution, and then to estimate the value of the unobservable quantity assigned to each member of the sample.

[121] *Algorithmic learning theory* (or *inductive inference*) is a framework for machine learning, introduced in E.M. Gold's seminal paper 'Language identification in the

Computational learning theory has led to practical algorithms. For example, PAC theory inspired boosting, VC theory led to *support vector machines*, and Bayesian inference led to Bayesian belief networks (see below).

---

limit' [Gol67]. The objective of language identification is for a machine running one program to be capable of developing another program by which any given sentence can be tested to determine whether it is 'grammatical' or 'ungrammatical'. The language being learned need not be English or any other natural language – in fact the definition of 'grammatical' can be absolutely anything known to the tester.

In the framework of algorithmic learning theory, the tester gives the learner an example sentence at each step, and the learner responds with a hypothesis, which is a suggested program to determine grammatical correctness. It is required of the tester that every possible sentence (grammatical or not) appears in the list eventually, but no particular order is required. It is required of the learner that at each step the hypothesis must be correct for all the sentences so far. A particular learner is said to be able to 'learn a language in the limit' if there is a certain number of steps beyond which its hypothesis no longer changes. At this point it has indeed learned the language, because every possible sentence appears somewhere in the sequence of inputs (past or future), and the hypothesis is correct for all inputs (past or future), so the hypothesis is correct for every sentence. The learner is not required to be able to tell when it has reached a correct hypothesis, all that is required is that it be true.

Gold showed that any language which is defined by a Turing machine program can be learned in the limit by another Turing–complete machine using enumeration. This is done by the learner testing all possible Turing machine programs in turn until one is found which is correct so far; this forms the hypothesis for the current step. Eventually, the correct program will be reached, after which the hypothesis will never change again (but note that the learner does not know that it won't need to change).

Gold also showed that if the learner is given only positive examples (that is, only grammatical sentences appear in the input, not ungrammatical sentences), then the language can only be guaranteed to be learned in the limit if there are only a finite number of possible sentences in the language (this is possible if, for example, sentences are known to be of limited length).

Language identification in the limit is a very theoretical model. It does not allow for limits of runtime or computer memory which can occur in practice, and the enumeration method may fail if there are errors in the input. However the framework is very powerful, because if these strict conditions are maintained, it allows the learning of any program known to be computable. This is because a Turing machine program can be written to mimic any program in any conventional programming language. Other frameworks of learning consider a much more restricted class of function than Turing machines, but complete the learning more quickly (in polynomial time). An example of such a framework is *probably approximately correct learning*.

Social and Emergent Learning

the *social and emergent learning* focuses on learning algorithms using the underlying concept of evolution, in other words, shaping a population $P(t)$ of candidate solutions x_i^t through the survival of the fittest members at time t. $P(t)$ is defined as:

$$P(t) = \{x_1^t, x_2^t, \ldots, x_n^t\}.$$

The attributes of a solution are represented with a particular pattern that is initialized by a *genetic algorithm*. As time passes, solution candidates are evaluated according to a specific fitness function that returns a measure of the candidate's fitness at that time. After evaluating all candidates the algorithm selects pairs for recombination. Genetic operators from each individual are used to produce new solutions that combine components of their parents. The fitness of a candidate determines the extent to which it reproduces. The general form of the genetic algorithm reads [Lug02]:

1. $t \leftarrow 0$;
2. Initialize population $P(t)$;
3. **while** termination condition not met **do**;
4. **for** each member x_i^t within $P(t)$ **do**;
5. $fitness(member) \leftarrow FitnessFunction(member)$;
6. **end for**;
7. select members from $P(t)$ based on $fitness(member)$;
8. produce offspring of selected members using generic operators;
9. replace members of $P(t)$ with offspring based on fitness;
10. $t \leftarrow t + 1$;
11. **end while**.

Reinforcement Learning

Recall that *reinforcement learning* (RL) is designed to allow computers to *learn by trial and error*. It is an approach to machine intelligence that combines two disciplines to solve a problem that each discipline cannot solve on its own. The first discipline, *dynamic programming* is a field in mathematics used to solve problems of optimization and control. The second discipline, supervised learning is discussed in section on neural networks below. In most real–life problems the correct answers required with supervised learning are not available, using RL the agent is simply provided with a *reward–signal* that implicitly trains the agent as required, Figure 1.3 illustrates the agent–environment interaction used with RL. The agent and the environment interact in a discrete sequence of time steps $t = 0, 1, 2, 3, ...$, for each time step the agent is presented with the current instance of the state $s_t \in S$ where S is the set of all possible states. The agent then uses the state to select and execute an action $a_t \in A(st)$ where $A(st)$ is the set of all possible actions available in state st. In the next time step the agent receives a reward $r_{t+1} \in R$, and

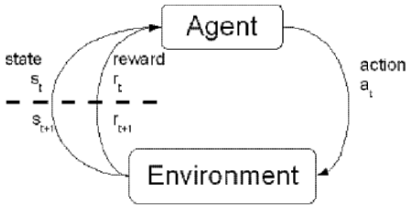

Fig. 1.3. The agent–environment interface in reinforcement learning (adapted from [SB98]).

is presented with a new state s_{t+1}. The system learns by mapping an action to each state for a particular environment. A specific mapping of actions and states is known as a *policy* π where $\pi_t(s, a)$ is the probability that $a_t = a$ if $s_t = s$. Actions available to agents can be separated into three different categories [SB98]:

- Low–level actions (e.g., supplying voltage to a motor);
- High–level actions (e.g., making a decision);
- Mental actions (e.g., shifting attention focus);

An important point to note is that according to Figure 1.3, the reward is calculated by the environment which is external to the agent. This is a confusing concept because at first it seems that the designer of an RL system is required to somehow implement something in the environment in order to provide an agent with appropriate rewards. The RL literature overcome this problem by explaining that the boundary between the agent and the environment need not be distinctively physical. The boundary of the agent is shortened to include only the reasoning process, everything outside the reasoning process which includes all other components of the agent, are treated as part of the environment. In the context of human reasoning, this is analogous to treating the human brain as the agent and the entire human body as part of the environment [Sio05].

Markov property of RL is concerned with the way that the state signal received from the environment is represented. This is an important issue when developing an RL system because all actions are directly dependent on the state of the environment. In a causal system the response of the environment for an action taken at time t will depend on all actions previously taken, formally written as

$$PR\{s_{t+1} = s', \quad r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, ..., s_0, a_0\}.$$

However, the state signal should not be expected to represent everything about the environment because certain information might be inaccessible or intentionally made unavailable.

When the response of the environment depends only on the state and action representations at time t, is it said to have the Markov property and

can be defined as

$$PR\{s_{t+1} = s', \quad r_{t+1} = r|s_t, a_t\}.$$

This means that the state signal is able to summarize all past sensations compactly such that all relevant information is retained for making decisions.

When a reinforcement learning problem satisfies the Markov property it is called a *Markov decision process* (MDP), additionally if the states and actions sets are finite then it is called a finite MDP. In some cases even when a particular problem is non–Markov it may be possible to consider it as an approximation of an MDP for the basis for learning, in such cases the learning performance will depend on how good the approximation is.

Reward function $R^a_{ss'}$ provides rewards depending on the actions of the agent. The sequence of rewards received after time step t is $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$, the agent learns by trying to maximize the sum of rewards received when starting from an initial state and proceeding to a terminal state. An additional concept is the one when an agent tries to maximize the expected discounted return as

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1},$$

where $0 \leq \gamma \leq 1$. This involves the agent discounting future rewards by a factor of γ.

There are two important classes of reward functions [HH97]. In the *pure delayed reward functions*, rewards are all zero except at a terminal state where the sign of the reward indicates whether it is a goal or penalty state. A classic example of pure delayed rewards is the cart–pole problem, where the cart is supporting a hinged inverted pendulum and the goal of the RL agent is to learn to balance the pendulum in an upright position. The agent has two actions in every state, move left and move right. The reinforcement function is zero everywhere except when the pole falls or the cart hits the end of the track, when the agent receives a -1 reward. Through such a set–up an agent will eventually learn to balance the pole and avoid the negative reinforcement. On the other hand, using the *minimum–time reward functions* it becomes possible to find the shortest path to a goal state. The reward function returns a reward of -1 for all actions except for the one leading to a terminal state for which the value is again dependent on whether it is a goal or penalty state. Due to the fact that the agent wants to maximize its rewards, it tries to achieve its goal at the minimum number of actions and therefore learns the optimal policy. An example used to illustrate this problem is driving a car up the hill problem, which is caused by the car not having enough thrust to drive up the hill on its own and therefore the RL agent needs to learn to use the momentum of the car climb the hill.

Value function. The issue of how an agent knows what is a good action is tackled using the *value function* $V^\pi(s)$ which provides a value of 'goodness' to states with respect to a specific policy. For MDPs, the information in a value function can be formally defined by

$$V^{\pi}(s) = E_{\pi}\{R_t|s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s\right\}$$

where $E_{\pi}\{\}$ denotes the expected value if the agent follows policy π, this is called the *state value function*. Similarly, the *action value function* starting from s, taking action a, and thereafter following policy π is defined by

$$Q^{\pi}(s,a) = E_{\pi}\{R_t|s_t = s,\, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s,\, a_t = a\right\}.$$

A value function that returns the highest value for the best action in each state is known as the *optimal value function*. $V^*(s)$ and $Q^*(s,a)$ denote the optimal state and action value functions and are given respectively by

$$V^*(s) = \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V^*(s')],$$
$$Q^*(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma \max_{a'} Q^*(s',a')].$$

Learning algorithms are concerned with how and when to update the value function using provided rewards. The differences in algorithms range depending on the required data that they need to operate, how they perform calculations and finally when this update takes place. Learning algorithms can be divided into three major classes: *dynamic programming*, *Monte–Carlo method* and *time–difference method*.

Dynamic programming (DP) works by assigning blame to the many decisions a system has to do while operating, this is done using two simple principles. Firstly, if an action causes something bad to happen immediately, then it learns not to do that action from that state again. Secondly, if all actions from a certain state lead to a bad result then that state should also be avoided. DP requires a *perfect environment model* in order to find a solution. Therefore the environment must have finite sets of states S and actions $A(s)$, and also finite sets of transition probabilities $P^a_{ss'} = \Pr\{s_{t+1} = s'|s_t = s,\, a_t = a\}$ and immediate rewards $R^a_{ss'} = E\{r_{t+1}|s_{t+1} = s',\, s_t = s,\, a_t = a\}$ for all $s \in S$, $a \in A(s)$. The value function in DP is updated using the equation

$$V^{\pi}(s) = \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V^*(s')].$$

Starting from the far right in this equation it can be seen that the reward received for taking an action is added to the discounted value of the resulting state of that action. However, a single action may have multiple effects in a complex environment leading to multiple resulting states. The value of each possible resulting state is multiplied by the corresponding transition probability and all results are added to get the actual value of a single action. In order

to calculate the value of the state itself, the value of each action is calculated and added to produce the full value of the state.

The two biggest problems encountered when developing applications using DP are [Sio05]: (i) the requirement of previously knowing all effects of actions taken in the environment, and (ii) the exponential increase in computation required to calculate the value of a state for only a small increase in possible actions and/or effects.

Monte Carlo (MC) methods however do not assume complete knowledge of the environment and require only experience through sampling sequences of states, actions and rewards from direct interaction with an environment. They are able to learn by segmenting sequences of actions into episodes and averaging rewards received as shown by the following algorithm [SB98]:

1: $\pi \longleftarrow$ policy to be evaluated;
2: $V \longleftarrow$ an arbitrary state–value function;
3: $Returns(s) \longleftarrow$ an empty list, for all $s \in S$;
4: **while** true **do;**
5: Generate an episode using;
6: **for** each state s appearing in the episode **do**;
7: $R \longleftarrow$ return following the first occurrence of s;
8: Append R to $Returns(s)$;
9: $V(s) \longleftarrow average(Returns(s))$;
10: **end for**;
11: **end while**;

Note that the algorithm requires the generation of an entire episode (line 5) before performing any updates to the value function.

MC is also able to estimate action values rather than state values, in this case policy evaluation is performed by estimating $Q^{\pi}(s, a)$, which is the expected return when starting in state s, taking action a, and thereafter following policy π. The relevant algorithm has the same structure as above. When MC is used for approximating optimal policies, the *generalized policy iteration* (GPI) is used. GPI maintains an approximate policy and an approximate value function, it then performs policy evaluation[122] and policy improvement[123] repeatedly. This means that the value function is updated to reflect the current policy while the policy is then improved with respect to the value function. Using these two processes GPI is able to maximize its rewards.

Temporal–Difference (TD) learning combines ideas from both MC and DP methods. Similarly to MC, TD methods are able to learn from experiences and do not need a model of the environment's dynamics. Like DP, TD methods update the value function based in part on estimates of future states

[122] Policy evaluation calculates the value function of a given policy.
[123] Policy improvement changes the policy such that it takes the best actions as dictated by the value function.

(this feature is called *bootstraping*) and hence do not require waiting for the episode to finish. An example of TD learning is the *Sarsa* algorithm [SB98]:

1: Initialize $Q(s, a)$ arbitrarily;
2: **for** each episode **do**;
3: Initialize s;
4: Choose a from s using policy *derived* from Q;
5: **for** each state s in episode **do**;
6: Take action a, observe r, s';
7: Choose a' from s' using policy *derived* from Q;
8: $Q(s, a) \longleftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$;
9: $s \longleftarrow s'$; $a \longleftarrow a'$;
10: **end for**;
11: **end for**;

The most important part of the algorithm is line 8 where the action value function is updated according to the rule:

$$Q(s, a) \longleftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)],$$

where α is called the step–size parameter and it controls how much the value function is changed with each update. Sarsa is an on–policy TD–algorithm and it requires the agent to select the following action before updating $Q(s, a)$. This is because $Q(s, a)$ is calculated by subtracting $Q(s, a)$ from the discounted value of $Q(s', a')$, which can only be known by selecting a'. Note that actions are selected using a policy that is based on the value function and in turn the value function is updated from the reward received.

Off–policy TD is able to approximate the optimal value function independently of the policy being followed. An example is the *Qlearning* algorithm [SB98]:

1: Initialize $Q(s, a)$ arbitrarily;
2: **for** each episode **do**;
3: Initialize s;
4: **for** Each state s in episode **do**;
5: Choose a' from s' using policy *derived* from Q;
6: Take action a, observe r, s';
7: $Q(s, a) \longleftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$;
8: $s \longleftarrow s'$;
9: **end for**;
10: **end for**;

The main difference between Sarsa and Qlearning lies in the calculation that updates the value function, the Qlearning update function is given by

$$Q(s, a) \longleftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)].$$

With Sarsa the value function is updated based on the *next chosen action*, while with Qlearning it is updated based on the *best known future action* even if that action is actually *not selected* in the next iteration of the algorithm.

Exploration versus exploitation. One of the more well known problems within the RL literature is the *exploration/exploitation problem*. During its operation the agent forms the *action estimates* $Q^{\pi}(a) = Q^{*}(a)$. The best known action at time t would therefore be

$$a_t^* = \arg\max_a Q_t(a).$$

An agent is said to be *exploring* when it tries an new action for a particular situation $a \neq a_t^*$. The reward obtained from the execution of that action is used to update the value function accordingly. An agent is said to be *exploiting* its learning knowledge when it chooses the *greedy action* (i.e., best action) indicated by its value function in a particular state $a = a_t^*$. In this case, the agent also updates the value function according to the reward received. This may have two effects, firstly, the reward may be similar to the one expected by the value function, which means that the value function is stabilizing on the problem trying to be solved. Secondly, it may be totally different to the value expected, therefore changing the value function and possibly the ordering of the actions with respect the their values. Hence, another action may subsequently become the 'best' action for that state.

An action selection policy controls the exploitation/exploration that is performed by the agent while learning. There are two types of policies commonly considered. Firstly, the *EGreedy policy* explores by selecting actions randomly but only for a defined percentage of all actions chosen as

$$a_t = \begin{cases} a_t^* & \text{if} \quad PR = (1-\epsilon), \\ \text{random} & \text{if} \quad PR = \epsilon. \end{cases}$$

For example, if $\epsilon = 0.1$ then the agent will explore only 10% of the time, the rest of the time it chooses the greedy action.

Secondly, the *SoftMax action selection* is more complex. It makes its choice based on the relation

$$a_t = \frac{\mathrm{e}^{Q_t(a)/\tau}}{\sum_{b=1}^{n} \mathrm{e}^{Q_t(a)/\tau}},$$

where τ is called the *temperature value*. A high temperature selects all actions randomly, while a low temperature selects actions in a greedy fashion. An intermediate temperature value causes SoftMax to select actions with a probability that is based on their value. This way actions with a high value have a greater chance of being selected while actions with a lower value have less chance of being selected. The advantage of SoftMax is that it tends to select the best action most of the time followed by the second–best, the third–best and so on, an action with a very low value is seldom executed. This is useful when a particular action is known to cause extremely bad rewards. Using SoftMax, that action will always get a very small probability of execution,

with EGreedy however, it has the same probability as any other action when exploring.

AI Programming Languages

Lisp

Recall that *Lisp* resps a family of computer programming languages with a long history and a distinctive fully–parenthesized syntax. Originally specified in 1958, Lisp is the second–oldest high–level programming language[124] in widespread use today; only Fortran is older. Like Fortran, Lisp has changed a great deal since its early days, and a number of dialects have existed over its history. Today, the most widely–known general–purpose Lisp dialects are Common Lisp[125] and Scheme.[126]

[124] Recall that a high–level programming language is a programming language that, in comparison to low–level programming languages, may be more abstract, easier to use, or more portable across platforms. Such languages often abstract away CPU operations such as memory access models and management of *scope*.

[125] Common Lisp, commonly abbreviated CL, is a dialect of the Lisp programming language, standardised by ANSI X3.226-1994. Developed to standardize the divergent variants of Lisp which predated it, it is not an implementation but rather a language specification. Several implementations of the Common Lisp standard are available, including commercial products and open source software.

Common Lisp is a general–purpose programming language, in contrast to Lisp variants such as Emacs Lisp and AutoLISP which are embedded extension languages in particular products. Unlike many earlier Lisps, Common Lisp (like Scheme) uses lexical variable scope.

Common Lisp is a multi–paradigm programming language that:

(i) Supports programming techniques such as imperative, functional and object-oriented programming.

(ii) Is dynamically typed, but with optional type declarations that can improve efficiency.

(iii) Is extensible through standard features such as Lisp macros (compile–time code rearrangement accomplished by the program itself) and reader macros (extension of syntax to give special meaning to characters reserved for users for this purpose).

[126] Scheme is a multi–paradigm programming language and a dialect of Lisp which supports functional and procedural programming. It was developed by Guy L. Steele and Gerald Jay Sussman in the 1970s. Scheme was introduced to the academic world via a series of papers now referred to as Sussman and Steele's Lambda Papers. There are two standards that define the Scheme language: the official IEEE standard, and a de facto standard called the Revisedn Report on the Algorithmic Language Scheme, nearly always abbreviated RnRS, where n is the number of the revision.

Scheme's philosophy is minimalist. Scheme provides as few primitive notions as possible, and, where practical, lets everything else be provided by

Lisp was originally created as a practical mathematical notation for computer programs, based on Church's[127] *lambda calculus* (which provides a theoretical framework for describing functions and their evaluation; though it is a mathematical abstraction rather than a programming language, lambda calculus forms the basis of almost all *functional programming languages*[128] today).

---

programming libraries. Scheme, like all Lisp dialects, has very little syntax compared to many other programming languages. There are no operator precedence rules because fully nested and parenthesized notation is used for all function calls, and so there are no ambiguities as are found in infix notation, which mimics conventional algebraic notation.

Scheme uses lists as the primary data structure, but also has support for vectors. Scheme was the first dialect of Lisp to choose static (a.k.a. lexical) over dynamic variable scope. It was also one of the first programming languages to support first–class continuations.

[127] Alonzo Church (June 14, 1903 — August 11, 1995) was an American mathematician and logician who was responsible for some of the foundations of theoretical computer science. Born in Washington, DC, he received a bachelor's degree from Princeton University in 1924, completing his Ph.D. there in 1927, under Oswald Veblen. After a postdoc at Göttingen, he taught at Princeton, 1929—1967, and at the University of California, Los Angeles, 1967–1990.

Church is best known for the following accomplishments:

(i) His proof that Peano arithmetic and first–order logic are undecidable. The latter result is known as *Church's theorem*.

(ii) His articulation of what has come to be known as *Church's thesis*.

(iii) He was the founding editor of the Journal of Symbolic Logic, editing its reviews section until 1979.

(iv) His creation of the *lambda calculus*.

The lambda calculus emerged in his famous 1936 paper showing the existence of an 'undecidable problem'. This result preempted Alan Turing's famous work on the halting problem which also demonstrated the existence of a problem unsolvable by mechanical means. He and Turing then showed that the lambda calculus and the Turing machine used in Turing's halting problem were equivalent in capabilities, and subsequently demonstrated a variety of alternative 'mechanical processes for computation'. This resulted in the *Church—Turing thesis*.

The lambda calculus influenced the design of the LISP programming language and functional programming languages in general. The Church encoding is named in his honor.

[128] Recall that *functional programming* is a programming paradigm that conceives computation as the evaluation of mathematical functions and avoids state and mutable data. Functional programming emphasizes the application of functions, in contrast with imperative programming, which emphasizes changes in state and the execution of sequential commands. A broader conception of functional programming simply defines a set of common concerns and themes rather than a list of distinctions from other paradigms. Often considered important are higher–order and first-class functions, closures, and recursion. Other common features of functional programming languages are continuations, *Hindley–Milner type inference systems*, non–strict evaluation, and monads.

Lisp quickly became the favored programming language for artificial intelligence research. As one of the earliest programming languages, Lisp pioneered many ideas in computer science, including tree data structures, automatic storage management, dynamic typing, object–oriented programming, and the self–hosting compiler.

The name Lisp derives from 'List Processing'. Linked lists are one of Lisp languages' major data structures, and Lisp source code is itself made up of lists. As a result, Lisp programs can manipulate source code as a data structure, giving rise to the macro systems that allow programmers to create new syntax or even new 'little languages' embedded in Lisp.

The interchangeability of code and data also give Lisp its instantly recognizable syntax. All program code is written as s–expressions, or parenthesized lists. A function call or syntactic form is written as a list with the function or operator's name first, and the arguments following: $(f\ x\ y\ z)$.

Lisp was invented by John McCarthy in 1958 while he was at MIT. McCarthy published its design in a paper in Communications of the ACM in 1960, entitled 'Recursive Functions of Symbolic Expressions and Their Computation by Machine'.[129] He showed that with a few simple operators and a notation for functions, one can build a Turing–complete language for algorithms. Lisp was first implemented by Steve Russell on an IBM 704 computer. Russell had read McCarthy's paper, and realized (to McCarthy's surprise) that the eval function could be implemented as a Lisp interpreter. The first complete Lisp compiler, written in Lisp, was implemented in 1962 by Tim Hart and Mike Levin at MIT. (AI Memo 39, 767 kB PDF.) This compiler introduced the Lisp model of incremental compilation, in which compiled and interpreted functions can intermix freely. The language used in Hart and Levin's memo is much closer to modern Lisp style than McCarthy's earlier code.

Largely because of its resource requirements with respect to early computing hardware (including early microprocessors), Lisp did not become as popular outside of the AI community as Fortran and the ALGOL–descended C language. Newer languages such as Java have incorporated some limited versions of some of the features of Lisp, but are necessarily unable to bring the coherence and synergy of the full concepts found in Lisp. Because of its suitability to ill–defined, complex, and dynamic applications, Lisp is presently enjoying some resurgence of popular interest.

Functional programming languages, especially 'purely functional' ones, have largely been emphasized in academia rather than in commercial software development. However, notable functional programming languages used in industry and commercial applications include Erlang (concurrent applications), R (statistics), Mathematica (symbolic math), J and K (financial analysis), and domain–specific programming languages like XSLT. Important influences on functional programming have been the *lambda calculus*, APL, Lisp and Haskell.

[129] McCarthy's original notation used bracketed 'M–expressions' that would be translated into S–expressions.

Prolog

Prolog is a *logic programming* language. The name Prolog is taken from 'programmation en logique' (which is French for 'programming in logic'). It was created by Alain Colmerauer and Robert Kowalski[130] around 1972 as an alternative to the American–dominated Lisp programming languages. It has been an attempt to make a programming language that enables the expression of logic instead of carefully specified instructions on the computer. In some ways Prolog is a subset of Planner, e.g., see Kowalski's early history of logic programming. The ideas in Planner were later further developed in the *Scientific Community Metaphor*.[131]

[130] Alain Colmerauer (born January 24, 1941) is a French computer scientist. He is the creator of the logic programming language Prolog and Q–Systems, one of the earliest linguistic formalisms used in the development of the TAUM–METEO machine translation prototype. He is a professor at the University of Aix–Marseilles, specialising in the field of constraint programming.

Robert Anthony Kowalski (born May 15, 1941 in Bridgeport, Connecticut, USA) is an American logician who has spent much of his career in the UK. He has been important in the development of logic programming, especially the programming language Prolog. He is also interested in legal reasoning.

[131] The Scientific Community Metaphor is one way of understanding scientific communities. In this approach, a high level programming language called Ether was developed that made use of pattern–directed invocation to invoke high–level procedural plans on the basis of messages (e.g. assertions and goals). The Scientific Community Metaphor builds on the philosophy, history and sociology of science with its analysis that scientific research depends critically on monotonicity, concurrency, commutativity, and pluralism to propose, modify, support, and oppose scientific methods, practices, and theories.

The first publications on the Scientific Community Metaphor (Kornfeld & Hewitt 1981, Kornfeld 1981, Kornfeld 1982) involved the development of a programming language named 'Ether' that invoked procedural plans to process goals and assertions concurrently by dynamically creating new rules during program execution. Ether also addressed issues of conflict and contradiction with multiple sources of knowledge and multiple viewpoints.

According to Carl Hewitt [Hew69], Scientific Community Metaphor systems have characteristics of:

(i) monotonicity (once something is published it cannot be withdrawn),

(ii) concurrency (scientists can work concurrently, overlapping in time and interacting with each other),

(iii) commutativity (publications can be read regardless of whether they initiate new research or become relevant to ongoing research),

(iv) pluralism (publications include heterogeneous, overlapping and possibly conflicting information),

(v) skepticism (great effort is expended to test and validate current information and replace it with better information), and

(vi) provenance (the provenance of information is carefully tracked and recorded).

Prolog is used in many AI programs and in *computational linguistics* (especially natural language processing, which it was originally designed for; the original goal was to provide a tool for computer–illiterate linguists) A lot of the research leading up to modern implementations of Prolog came from spin–off effects caused by the *fifth generation computer systems* project (FGCS) which chose to use a variant of Prolog named *Kernel Language* for their operating system (however, this area of research is now actually almost defunct).

Prolog is based on *first–order predicate calculus*;[132] however it is restricted to allow only *Horn clauses*.[133] Execution of a Prolog program is effectively an application of theorem proving by *first–order resolution*.

'Planner' is a programming language designed by Carl Hewitt at MIT, and first published in 1969. First subsets such as Micro–Planner and Pico–Planner were implemented and then essentially the whole language was implemented in Popler and derivations such as QA–4, Conniver, QLISP and Ether.

[132] Recall that *predicate calculus* consists of

1. *formation rules* (i.e. recursive definitions for forming well–formed formulas),
2. *transformation rules* (i.e. inference rules for deriving theorems), and
3. *axioms* or *axiom schemata* (possibly a countably infinite number).

When the set of axioms is infinite, it is required that there be an algorithm which can decide for a given well–formed formula, whether it is an axiom or not. There should also be an algorithm which can decide whether a given application of an inference rule is correct or not.

[133] A Horn clause is a clause (a disjunction of literals) with at most one positive literal. A Horn clause with exactly one positive literal is a definite clause; a Horn clause with no positive literals is sometimes called a goal clause, especially in logic programming. A Horn formula is a conjunctive normal form formula whose clauses are all Horn; in other words, it is a conjunction of Horn clauses. A dual–Horn clause is a clause with at most one negative literal. Horn clauses play a basic role in logic programming and are important for constructive logic. For example,

$$\neg p \vee \neg q \vee \cdots \vee \neg t \vee u$$

is a definite Horn clause. Such a formula can be rewritten in the following form, which is more common in logic programming,

$$(p \wedge q \wedge \cdots \wedge t) \rightarrow u.$$

The relevance of Horn clauses to theorem proving by *first–order resolution* is that the resolution of two Horn clauses is a Horn clause. Moreover, the resolution of a goal clause and a definite clause is again a goal clause. In automated theorem proving, this can lead to greater efficiencies in proving a theorem (represented as a goal clause). Prolog is a programming language based on Horn clauses. Horn clauses are also of interest in computational complexity, where the problem of finding a set of variable assignments to make a conjunction of Horn clauses true is a *P–complete problem*.

Recall that a *resolution rule* in *propositional logic* is a single valid inference rule producing, from two clauses, a new clause implied by them. The resolution rule takes two clauses – a clause is a disjunction of literals – containing complementary literals, and produces a new clause with all the literals of both except for the complementary ones. The clause produced by the resolution rule is called the resolvent of the two input clauses. When the two clauses contain more than one pair of complementary literals, the resolution rule can be applied (independently) for each such pair. However, only the pair of literals that are resolved upon can be removed: all other pair of literals remain in the resolvent clause.

When coupled with a complete *search algorithm*, the resolution rule yields a sound and complete algorithm for deciding the *satisfiability* of a propositional formula, and, by extension, the validity of a sentence under a set of axioms. This resolution technique uses *proof by contradiction* and is based on the fact that any sentence in propositional logic can be transformed into an equivalent sentence in *conjunctive normal form*. Its steps are:

1. All sentences in the knowledge base and the negation of the sentence to be proved (the conjecture) are conjunctively connected.
2. The resulting sentence is transformed into a conjunctive normal form (treated as a set of clauses, S).
3. The resolution rule is applied to all possible pairs of clauses that contains complementary literals. After each application of the resolution rule, the resulting sentence is simplified by removing repeated literals. If the sentence contains complementary literals, it is discarded (as a *tautology*). If not, and if it is not yet present in the clause set S, it is added to S, and is considered for further resolution inferences.
4. If after applying a resolution rule the empty clause is derived, the complete formula is unsatisfiable (or contradictory), and hence it can be concluded that the initial conjecture follows from the axioms.
5. If, on the other hand, the empty clause cannot be derived, and the resolution rule cannot be applied to derive any more new clauses, the conjecture is not a theorem of the original knowledge base.

In first order logic resolution condenses the traditional syllogisms of logical inference down to a single rule.

Fundamental Prolog concepts are *unification*, *tail recursion*, and *backtracking* (a strategy for finding solutions to *constraint satisfaction problems*). The concept of unification is one of the main ideas behind Prolog. It represents the mechanism of binding the contents of variables and can be viewed as a kind of one–time assignment. In Prolog, this operation is denoted by symbol '='. In traditional Prolog, a variable X which is uninstantiated, i.e., no previous unifications were performed on it, can be unified with an atom, a term, or another uninstantiated variable, thus effectively becoming its alias. In many modern Prolog dialects and in first–order logic calculi, a variable cannot be unified with a term that contains it; this is the so called 'occurs check'.

A *Prolog atom* can be unified only with the same atom. Similarly, a *Prolog term* can be unified with another term if the top function symbols and arities of the terms are identical and if the parameters can be unified simultaneously (note that this is a *recursive behaviour*). Due to its declarative nature, the order in a sequence of unifications is (usually) unimportant [BS01].

The tail recursion (or *tail–end recursion*) is a special case of recursion that can be easily transformed into an *iteration*. Such a transformation is possible if the recursive call is the last thing that happens in a function. Replacing recursion with iteration, either manually or automatically, can drastically decrease the amount of stack space used and improve efficiency. This technique is commonly used with functional programming languages, where the declarative approach and explicit handling of state promote the use of recursive functions that would otherwise rapidly fill the call stack.

Prolog has a built in mechanism for parsing *context–free grammar* (CFG), a formal grammar in which every *production rule* is of the form: $V \rightarrow w$, where V is a non–terminal symbol and w is a string consisting of terminals and/or non–terminals. The term 'context–free' comes from the fact that the non–terminal V can always be replaced by w, regardless of the context in which it occurs. A formal language is context–free if there is a context–free grammar that generates it.

Context–free grammars are powerful enough to describe the syntax of most programming languages; in fact, the syntax of most programming languages are specified using context–free grammars. On the other hand, context–free grammars are simple enough to allow the construction of efficient parsing algorithms which, for a given string, determine whether and how it can be generated from the grammar. The metasyntax called *Backus–Naur form* (BNF), is the most common notation used to express context–free grammars.

ACT–R: Combining Natural and Computational Intelligence

ACT–R (Adaptive Control of Thought–Rational) is a cognitive architecture mainly developed by John Anderson[134] at the Carnegie Mellon University (see [And83, And80, And90]). Like any cognitive architecture, ACT–R aims to define the basic and irreducible basic cognitive and perceptual operations that enable the human mind. In theory, each task that humans can perform should consist of a series of these discrete operations. Most of the ACT–R basic

[134] John Robert Anderson (born 1947 in Vancouver, British Columbia) is a professor of psychology and computer science at Carnegie Mellon University. He is widely known for his cognitive architecture ACT–R [And84]. He has published many papers on cognitive psychology, served as present of the Cognitive Science Society, and received many scientific awards, including one from the American Academy of Arts and Sciences. He is a fellow of the National Academy of Sciences. Anderson was an early leader in research on intelligent tutoring systems, and many of Anderson's former students, such as Kenneth Koedinger and Neil Heffernan, have become leaders in that area.

assumptions are also inspired by the progresses of cognitive neuroscience, and, in fact, ACT–R can be seen and described as way of specifying how the brain itself is organized in a way that enables individual processing modules to produce cognition.

Like other influential cognitive architectures (including Soar and EPIC), the ACT–R theory has a computational implementation as an interpreter of a special coding language. The interpreter itself is written in Lisp, and might be loaded into any of the most common distributions of the Lisp language. This enables researchers to specify models of human cognition in the form of a script in the ACT–R language. The language primitives and data–types are designed to reflect the theoretical assumptions about human cognition. These assumptions are based on numerous facts derived from experiments in cognitive psychology and brain imaging.

In recent years, ACT–R has also been extended to make quantitative predictions of patterns of activation in the brain, as detected in experiments with fMRI. In particular, ACT–R has been augmented to predict the exact shape and time–course of the BOLD response of several brain areas, including the hand and mouth areas in the motor cortex, the left prefrontal cortex, the anterior cingulate cortex, and the basal ganglia.

ACT–R's most important assumption is that human knowledge can be divided into two irreducible kinds of representations: declarative and procedural. Within the ACT–R code, declarative knowledge is represented in form of chunks, i.e., vector representations of individual properties, each of them accessible from a labelled slot. On the other hand, chunks are held and made accessible through buffers, which are the front–end of what are modules, i.e. specialized and largely independent brain structures.

There are two types of modules:

1. Perceptual–motor modules, which take care of the interface with the real world (i.e., with a simulation of the real world). The most well–developed perceptual–motor modules in ACT–R are the visual and the manual modules.
2. Memory modules. There are two kinds of memory modules in ACT–R: (i) Declarative memory, consisting of facts such as Washington, D.C. is the capital of United States, France is a country in Europe, or $2 + 3 = 5$; and (ii) Procedural memory, made of productions. Productions represent knowledge about how we do things: for instance, knowledge about how to type the letter 'Q' on a keyboard, about how to drive, or about how to perform addition.

Over the years, ACT–R models has been used in more than 500 different scientific publications, and has been cited in a huge amount of others. It has been applied in the following areas:

1. Learning and Memory
2. Higher level cognition, Problem solving and Decision making

3. Natural language, including syntactic parsing, semantic processing and language generation
4. Perception and Attention

More recently, more than two dozen papers made use of ACT–R for predicting brain activation patterns during imaging experiments, and it has also been tentatively used to model neuropsychological impairments and mental disorders.

Beside its scientific application in cognitive psychology, ACT–R used in other, more application–oriented domains.

1. Human–computer interaction to produce user models that can assess different computer interfaces,
2. Education, where ACT–R–based cognitive tutoring systems try to 'guess' the difficulties that students may have and provide focused help
3. Computer–generated forces to provide cognitive agents that inhabit training environments

Some of the most successful applications, the Cognitive Tutors for Mathematics, are used in thousands of schools across the United States. Such 'Cognitive Tutors' are being used as a platform for research on learning and cognitive modelling as part of the Pittsburgh Science of Learning Center.

After the publication of 'The Atomic Components of Thought' [And90], Anderson become more and more interested in the underlying neural plausibility of his life–time theory, and began to use brain imaging techniques pursuing his own goal of understanding the computational underpinnings of human mind. The necessity of accounting for brain localization pushed for a major revision of the theory. ACT–R 5.0, presented in 2002, introduced the concept of modules, specialized sets of procedural and declarative representations that could be mapped to known brain systems. In addition, the interaction between procedural and declarative knowledge was mediated by newly introduced buffers, specialized structures for holding temporarily active information (see the section above). Buffers were thought to reflect cortical activity, and a subsequent series of studies later confirmed that activations in cortical regions could be successfully related to computational operations over buffers. The theory was first described in the 2004 paper 'An Integrated Theory of Mind' [ABB04]. No major changes have occurred since then in the theory, but a new version of the code, completely rewritten, was presented in 2005 as ACT–R 6.0. It also included significant improvements in the ACT–R coding language.

Facial Recognition and Biometrics

A Facial Recognition (FR) system is a computer–driven application for automatically identifying a person from a digital image. It does that by comparing selected facial features in the live image and a facial database. It is typically

used for security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

Popular FR algorithms include *eigenfaces*, the *Hidden Markov models*, and the neuronal motivated *dynamic link matching*. A newly emerging trend, claimed to achieve previously unseen accuracies, is 3D face recognition. Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. Tests on the FERET database, the widely used industry benchmark, showed that this approach is substantially more reliable than previous algorithms.

FR is based on the computer identification of unknown face images by comparison with a single known image or database of known images. A FR may be used for access control (one–to–one) or for surveillance of crowds to locate people of interest (one–to–many or many–to–many). Access control FRs are often used in highly controlled environments, which means that the input data is of predictable quality, resulting in relatively high levels of performance. Surveillance applications (which are often covert), may call for a large number of faces to be compared with a large stored database of images to determine if there are any matches. This can result in a large number of false alarms. In addition, due to the nature of the surveillance application, the images obtained are often of poor quality, since it is often difficult to adequately control all the environmental conditions. This can reduce the ability of the FR to find a correct match with an enrolled image.

Modes of Operation

FR systems have two functional modes: enrolment and operation. Each mode used the same signal processing approach to extract salient information from the sensor data. In the enrolment phase, face data on known subjects is extracted and stored in a database of known persons (often called the 'gallery'). In general, each individual is sampled a number of times during enrolment, to ensure that the stored data is truly representative of that individual.

Once a database of known subjects is enrolled, the system may be used in the operational mode. In this mode, data from people who are not yet identified are processed in the same way as the enrolment data and the salient features are compared with the database to see if there is a match. When the degree of match is above some form of threshold, an action is generally required. A key to effective operation of an FRS is the image processing that extracts the salient features of faces for comparison with stored data.

Signal Processing Operations

The signal processing operations typically involved in FR include those listed below, either as discrete operations (an algorithmic approach) or in combination (e.g., a neural network approach):

(i) Face Capture: The first stage in the FR process is to identify objects that could be faces and then discard the rest of the scene. The face capture process could be as simple as a blob detector that sorts on size and

shape, or it may include higher level processes that look for features such as eye/nose/mouth geometry, color information or motion and location to identify objects that are face–like.

(ii) Normalization: Once faces have been identified they must be presented to the classifier in a form that compensates for variability in brightness/colour due to lighting, camera and frame grabber characteristics, as well as geometric distortions due to distance, pose and viewing aspect angles. Typical intensity normalization may involve grey scale modification of regions of interest to provide fixed average levels and contrast. Scale errors may be minimized by re-sampling the faces to produce constant size inputs to following stages. In general, the distance between eye pupils is used as the baseline measure to re–scale images and it is critical that this parameter be accurately determined, either by the software or manually.

(iii) Feature Extraction: Feature extraction is the process that takes the normalized version of each real–world face image and generates a compact data vector that uniquely describes it for use by the classification/database engine.

(iv) Database Comparison: Unknown subjects and a target sample are compared with the known database. Face images are gathered using the same (or a similar) sensor as was used for enrolment and this data is processed in the same way as the enrolment data. Following salient feature extraction, the incoming data vector is compared with each template in the database to determine the goodness of match with known data and a match measure is generated for each comparison.

(v) Decision and Action: A decision making process follows the match measurement, whereby the outcome is declared to be either a true match or a non–match, based on the match measure. This decision may be made by comparing the match value to a threshold setting. Any match measure that is on higher side of the threshold is declared to be a true match and any on the other is a non-match. The process of facial recognition is complex and many of the processes outlined above are highly dependent upon external variables. This can lead to considerable difficulty in the evaluation of the technologies involved.

Evaluation Methods

Phillips *et al.* [PMW00] have given a general introduction to evaluating biometric systems. They focused on biometric applications that give the user some control over data acquisition. These applications recognize subjects from mug shots, passport photos and scanned fingerprints. They concentrated on two major kinds of biometric systems: identification and verification. In identification systems, a biometric signature of an unknown person is presented to a system. The system compares the new biometric signature with a database of biometric signatures of known individuals. On the basis of the comparison, the system reports (or estimates) the identity of the unknown person from

this database. Systems that rely on identification include those that check for multiple applications by the same person for welfare benefits and driver's licences.

In verification systems, a user presents a biometric signature and a claim that a particular identity belonged to the biometric signature. The algorithm either accepts or rejects the claim. Alternatively, the algorithm could return a confidence measurement of the claim's validity. Verification applications include those that authenticate identity during point-of-sale transactions or that control access to computers or secure buildings.

Performance statistics for verification applications differ substantially from those for identification systems. The main performance measure for an identification system is that system's ability to identify the owner of a biometric signature. More specifically, the performance measure is equal to the percentage of queries in which the correct answer could be found in the top few matches.

Mansfield and Wayman [MW02] elaborated best practice in testing and reporting the performance of biometric devices. The purpose of their report, which is a revision of their original version [MW00], was to summarize the current understanding by the biometrics community of the best scientific practices for conducting technical performance testing toward the end of field performance estimation. The aims of the authors were as follows:

(1) To provide a framework for developing and fully describing test protocols.
(2) To help avoid systematic bias due to incorrect data collection or analytic procedures in evaluations.
(3) To help testers achieve the best possible estimate of field performance while expending minimum effort in conducting their evaluation.
(4) To improve understanding of the limits of applicability of test results and test methods.

The recommendations in this paper were extremely general in nature. It was noted that it might not be possible to follow best practice completely in any test. Compromises often need to be made. In such situations the experimenter has to decide on the best compromise to achieve the evaluation objectives, but should also report what has been done to enable a correct interpretation to be made of the results.

The FERET Program

The Face Recognition Technology (FERET) program, which was sponsored by the Department of Defense (DoD) Counterdrug Technology Program, commenced in September 1993. The primary mission of the FERET program was to develop automatic face recognition capabilities that could be employed to assist security, intelligence and law enforcement personnel in the performance of their duties.

The FERET program initially consisted of three one year phases. The objective of the first phase was to establish the viability of automatic face

recognition algorithms, and to determine a performance baseline against which to measure future progress. The goals of the other two phases was to further develop face recognition technology. After the completion of phase 2 the FERET demonstration effort was commenced, with the goals to port FERET evaluated algorithms to real–time experimental/demonstration systems.

The program focused on three major areas:

1. Sponsoring Research: The goal of the sponsored research was to develop facial recognition algorithms. After a broad agency announcement for algorithm development proposals, twenty–four submissions were received and evaluated by DoD and law enforcement personnel. Five contracts were initially awarded, and three of these teams were selected to continue their development for phase 2.
2. Collecting the FERET database: The FERET database of facial images was a vital part of the overall FERET program and promised to be key to future work in face recognition, because it provided a standard database for algorithm development, test and evaluation, and most importantly, the images were gathered independently from the algorithm developers. The images were collected in a semi–controlled environment, with the same physical setup used in each photography session to maintain a degree of consistency throughout the database. However, because the equipment had to be reassembled for each session, there was some minor variation in images collected on different dates. The FERET database was collected in 15 sessions between August 1993 and July 1996. The database contains 1564 sets of images for a total of 14,126 images that includes 1199 individuals and 365 duplicate sets of images. A duplicate set is a second set of images of a person already in the database and was usually taken on a different day.
3. Performing the FERET evaluations: Before the FERET database was created, a large number of papers reported outstanding recognition results (usually $>95\%$ correct recognition) on limited–size databases (usually <50 individuals). Only a few of these algorithms reported results on images utilizing a common database – the FERET database made it possible for researchers to develop algorithms on a common database and to report results in the literature using this database. More importantly, the FERET database and evaluations clarified the state of the art in face recognition and pointed out general directions for future research. Three sets of evaluations were performed, with the last two evaluations being administered multiple times. The first FERET evaluation took place in August 1994, the Aug94 evaluation. This evaluation was designed to measure performance on algorithms that could automatically locate, normalize, and identify faces from a database. The test consisted of three subtests, each with a different gallery and probe set. The first subtest examined the ability of algorithms to recognize faces from a gallery of 316 individuals. The second subtest was the false-alarm test, which measured how well an algorithm

rejects faces not in the gallery. The third subtest baselined the effects of pose changes on performance. The second FERET evaluation took place in March of 1995, the Mar95 evaluation. The goal was to measure progress since the initial FERET evaluation, and to evaluate these algorithms on larger galleries (817 individuals). An added emphasis of this evaluation was on probe sets that contained duplicate images, where a duplicate image was defined as an image of a person whose corresponding gallery image was taken on a different date. The third FERET evaluations took place in September of 1996, the Sep96 evaluation. For the Sept96 evaluation, a new evaluation protocol was designed which required algorithms to match a set of 3323 images against a set of 3816 images. The new protocol design allowed the determination of performance scores for multiple galleries and probe sets, and perform a more detailed performance analysis. There were two versions of the September 1996 evaluation. The first tested partially automatic algorithms by providing the images with the coordinates of the center of the eyes. The second tested fully automatic algorithms by providing the images only.
Further details on the methodology of the FERET program can be found in [PMW00].

Eigenfaces

Recall that *eigenfaces* are a set of *eigenvectors*[135] used in the computer vision problem of human FR. These eigenvectors are derived from the *covariance matrix* of the *probability distribution* of the high–dimensional vector space of possible human faces, in a similar fashion as in *factor analysis* described above. Many authors prefer the term eigenimage rather than eigenface, as the technique has been used for handwriting, lip reading, voice recognition, and medical imaging.

In layman's terms, eigenfaces are a set of 'standardized face ingredients', derived from multivariate correlation analysis of many pictures of faces. Any human face can be considered to be a combination of these standard faces. One person's face might be made up of 10% from face 1, 24% from face 2

[135] Recall that an *eigenvector* of a transformation is a non–null vector whose direction is unchanged by that transformation. The factor by which the magnitude is scaled is called the *eigenvalue* of that vector. Often, a transformation is completely described by its eigenvalues and eigenvectors. An *eigenspace* is a set of eigenvectors with a common eigenvalue. These concepts play a major role in several branches of both pure and applied mathematics — appearing prominently in linear algebra, functional analysis, and even a variety of nonlinear situations.

It is common to prefix any natural name for the solution with eigen instead of saying eigenvector. For example, *eigenfunction* if the eigenvector is a function, eigenmode if the eigenvector is a harmonic mode, eigenstate if the eigenvector is a quantum state, and so on (e.g. the eigenface example below). Similarly for the eigenvalue, e.g., *eigenfrequency* if the eigenvalue is (or determines) a frequency.

and so on. This means that if we want to record someone's face for use by FR software, we can use far less space than would be taken up by a digitised photograph.

To generate a set of eigenfaces, a large set of digitized images of human faces, taken under the same lighting conditions, are normalized to line up the eyes and mouths. They are then all resampled at the same pixel resolution (say $m \times n$), and then treated as mnD vectors whose components are the values of their pixels. The eigenvectors of the covariance matrix of the statistical distribution of face image vectors are then extracted. It should be noted that these are the same as the eigenvectors from principal components analysis (PCA, see above), the statistical method from which eigenimaging is derived. Since the eigenvectors belong to the same vector space as face images, they can be viewed as if they were $m \times n$ pixel face images: hence the name eigenfaces. Viewed in this way, the principal eigenface looks like a bland androgynous average human face. Some subsequent eigenfaces can be seen to correspond to generalized features such as left–right and top–bottom asymmetry, or the presence or absence of a beard. Other eigenfaces are hard to categorize, and look rather strange.

When properly weighted, eigenfaces can be summed together to create an approximate gray–scale rendering of a human face. Remarkably few eigenvector terms are needed to give a fair likeness of most people's faces, so eigenfaces provide a means of applying data compression to faces for identification purposes (see, e.g., [Abd88]).

Dynamic Link Matching

The *dynamic link matching* (DLM) is a neural FR–system based on the Gabor–wavelet transform [Mal85, Mal88, KMM94, LVB93, Wis95]. The system is inherently invariant with respect to shift, and is robust against many other variations, most notably rotation in depth and deformation. The system consists of an image domain and a model domain, which is tentatively identified with primary visual cortex and infero–temporal cortex. Both domains have the form of neural sheets of hypercolumns, which are composed of simple feature detectors (modeled as Gabor wavelets). Each object is represented in memory by a separate model sheet, that is, a 2D array of features. The match of the image to the models is performed by network self–organization, in which rapid reversible synaptic plasticity of the connections ('dynamic links') between the two domains is controlled by signal correlations, which are shaped by fixed inter–columnar connections and by the dynamic links themselves. The system requires very little genetic or learned structure, relying essentially on the rules of rapid synaptic plasticity and the a priori constraint of preservation of topography to find matches. This constraint is encoded within the neural sheets with the help of lateral connections, which are excitatory over short range and inhibitory over long range.

Topographical relationships between nodes in the DLM–system are encoded by excitatory and inhibitory lateral connections (see Figure 1.4). The

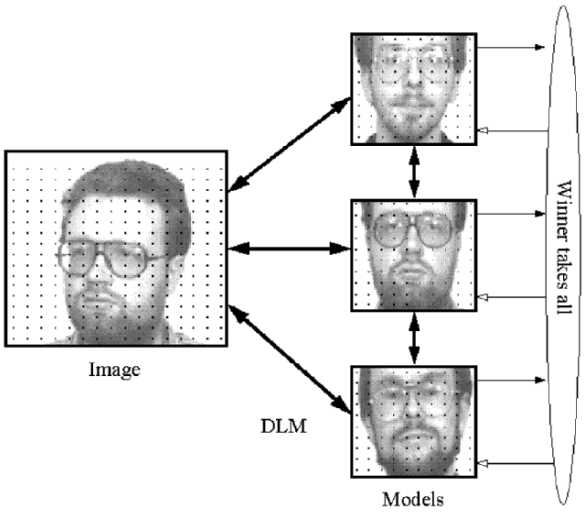

Fig. 1.4. Architecture of the DLM face recognition system. Several models are stored as neural layers of local features on a 1010 grid, as indicated by the black dots. A new image is represented by a 1617 layer of nodes. Initially, the image is connected all–to–all with the models. The task of DLM is to find the correct mapping between the image and the models, providing translational invariance and robustness against distortion. Once the correct mapping is found, a simple winner–take–all mechanism can detect the model that is most active and most similar to the image (adapted from [Mal85, Mal88, KMM94, LVB93, Wis95]).

model graphs are scaled horizontally and vertically and aligned manually, such that certain nodes of the graphs are placed on the eyes and the mouth. Model layers (1010 neurons) are smaller than the image layer (1617 neurons). Since the face in the image may be arbitrarily translated, the connectivity between model and image domain has to be all–to–all initially. The connectivity matrices are initialized using the similarities between the jets of the connected neurons. DLM serves as a process to restructure the connectivity matrices and to find the correct mapping between the models and the image. The models cooperate with the image depending on their similarity. A simple winner–take–all mechanism sequentially rules out the least active and least similar models, and the best–fitting one eventually survives.

Face Recognition Vendor Tests

Face Recognition Vendor Tests (FRVT) provide independent government evaluations of commercially available and mature prototype face recognition systems. During the FERET program face recognition technology was primarily found in prototype systems in universities and research labs. By 2000 systems were available on the commercial market, so FRVT 2000 was instigated to evaluate the capabilities of these commercial systems. Sponsored by the

Defense Advanced Research Projects Agency (DARPA), DoD Counterdrug Technology Development Program Office and National Institute of Justice (NIJ), and designed by the National Institute of Standards and Technology (NIST) the FRVT 2000 was based on the FERET evaluations and the evaluation methodology philosophy outlined by [PMW00].

The FRVT 2000 was a technology evaluation consisting of two components: the Recognition Performance Test and the Product Usability Test. The goal of the Recognition Performance Test was to compare competing techniques for performing facial recognition, with all systems tested on a standardized database. The product usability test examined system properties for performing access control. Five commercial products were evaluated, and the results of the tests can be found at get references from Lit Review Report. Under the USA Patriot Act, NIST is mandated to measure the accuracy of biometric technologies. In accordance with this legislation, NIST, in cooperation with other Government agencies, is conducting the Face Recognition Vendor Test 2002 FRVT 2002. Now sponsored or supported by 16 organisations, including some non–US agencies, the FRVT 2002 aims to assess the state–of–the–art in face recognition technology, and is conducting a technology evaluation of both mature prototype and commercially available systems face recognition systems.

Hidden Markov Models

A *hidden Markov model* (HMM) is a statistical model where the system being modelled is assumed to be a *Markov process*[136] with unknown parameters, and the challenge is to determine the hidden parameters from the observable

[136] Recall that a *Markov process* is a stochastic process that has a *Markov property*, or *Markov assumption*. Technically, there are three well–known special cases of the *Chapman–Kolmogorov equation*, describing a general Markov process (see [Gar85]):

1. When both $B_{ij}[x(t),t]$ and $W(t)$ are zero, i.e., in the case of pure deterministic motion, it reduces to the *Liouville equation*

$$\partial_t P(x',t'|x'',t'') = -\sum_i \frac{\partial}{\partial x^i}\left\{A_i[x(t),t]\,P(x',t'|x'',t'')\right\}.$$

2. When only $W(t)$ is zero, it reduces to the *Fokker–Planck diffusion equation*

$$\partial_t P(x',t'|x'',t'') = -\sum_i \frac{\partial}{\partial x^i}\left\{A_i[x(t),t]\,P(x',t'|x'',t'')\right\} + \frac{1}{2}\sum_{ij}\frac{\partial^2}{\partial x^i \partial x^j}\left\{B_{ij}[x(t),t]\,P(x',t'|x'',t'')\right\}.$$

3. When both $A_i[x(t),t]$ and $B_{ij}[x(t),t]$ are zero, i.e., the state–space consists of integers only, it reduces to the *Master equation* of discontinuous jumps

parameters.[137] The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. A HMM can be considered as the simplest *dynamic Bayesian network*.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

The HMM–architecture is depicted in Figure 1.5. From this diagram, it is clear that the value of the *hidden variable* $x(t)$ (at time t) only depends on the value of the hidden variable $x(t-1)$ (at time $t-1$). Similarly, the value of the *observed variable* $y(t)$ only depends on the value of the hidden variable $x(t)$ (both at time t).

The probability of observing a sequence $Y = y(0), y(1), \ldots, y(L-1)$ of length L in HMM is given by:

$$P(Y) = \sum_X P(Y \mid X) P(X),$$

$$\partial_t P(x', t'|x'', t'') = \int dx \left\{ W(x'|x'', t)\, P(x', t'|x'', t'') - W(x''|x', t)\, P(x', t'|x'', t'') \right\}.$$

The *Markov assumption* can now be formulated in terms of the conditional probabilities $P(x^i, t_i)$: if the times t_i increase from right to left, the conditional probability is determined entirely by the knowledge of the most recent condition. Markov process is generated by a set of conditional probabilities whose probability–density $P = P(x', t'|x'', t'')$ evolution obeys the general *Chapman–Kolmogorov integro–differential equation*

$$\begin{aligned} \partial_t P = & -\sum_i \frac{\partial}{\partial x^i} \{A_i[x(t), t]\, P\} \\ & + \frac{1}{2} \sum_{ij} \frac{\partial^2}{\partial x^i} \partial x^j \{B_{ij}[x(t), t]\, P\} \\ & + \int dx \left\{ W(x'|x'', t)\, P - W(x''|x', t)\, P \right\} \end{aligned}$$

including: *deterministic drift*, *diffusion fluctuations* and *discontinuous jumps* (given respectively in the first, second and third rows).

[137] Hidden Markov Models were first described in a series of statistical papers by Leonard Baum in the second half of the 1960s. One of the first applications of HMMs was speech recognition, starting in the mid–1970s. In the second half of the 1980s, HMMs began to be applied to the analysis of biological sequences, in particular DNA. Since then, they have become ubiquitous in the field of bioinformatics.

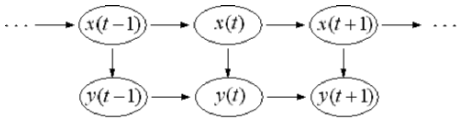

Fig. 1.5. Generic architecture of a Hidden Markov model. Each oval shape represents a random variable that can adopt a number of values. The random variable $x(t)$ is the value of the *hidden variable* at time t. The random variable $y(t)$ is the value of the *observed variable* at time t. The arrows in the diagram denote *conditional dependencies*.

where the sum runs over all possible hidden node sequences $X = x(0), x(1), \ldots, x(L-1)$. A *brute force* calculation of $P(Y)$ is intractable for realistic problems, as the number of possible hidden node sequences typically is extremely high. The calculation can however be speeded up enormously using a *dynamic programming* algorithm, called the *forward algorithm*.

Recall that dynamic programming, invented by Richard Bellman,[138] is a method for reducing the runtime of algorithms exhibiting the properties of:

1. Overlapping subproblems (the problem can be broken down into subproblems which are reused several times),[139]
2. Optimal substructure (optimal solution can be constructed efficiently from optimal solutions to its subproblems; used to determine the usefulness of dynamic programming and *greedy algorithms*[140] in a problem), and

[138] Richard Ernest Bellman (1920–1984) was an applied mathematician, celebrated for his invention of *dynamic programming* in 1953, and important contributions in other fields of mathematics, including the *Bellman equation* and *Hamilton–Jacobi–Bellman equation*.

A well–known term in computation coined by Bellman is *curse of dimensionality*: the problem caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space (e.g., 'rules explosion' in fuzzy logic systems). Similarly, the curse of dimensionality is a significant obstacle in machine learning problems that involve learning from few data samples in a high–dimensional feature space.

[139] For example, the problem of computing the *Fibonacci sequence* exhibits *overlapping subproblems*. The problem of computing the nth Fibonacci number, $F(n)$, can be broken down into the subproblems of computing $F(n-1)$ and $F(n-2)$, and then adding the two. The subproblem of computing $F(n-1)$ can itself be broken down into a subproblem that involves computing $F(n-2)$. Therefore the computation of $F(n-2)$ is reused, and the *Fibonacci sequence* thus exhibits overlapping subproblems.

[140] A *greedy algorithm* is an algorithm that follows the problem solving metaheuristic of making the locally optimum choice at each stage with the hope of finding the global optimum. For instance, applying the greedy strategy to the *traveling salesman problem* yields the following algorithm: 'At each stage visit the unvisited city nearest to the current city'. In general, greedy algorithms have five pillars: (i) a candidate set, from which a solution is created; (ii) a selection function,

3. Memoization (speeding up programs by storing the results of functions for later reuse, rather than recomputing them).[141]

---

which chooses the best candidate to be added to the solution; (iii) a feasibility function, that is used to determine if a candidate can be used to contribute to a solution; (iv) an objective function, which assigns a value to a solution, or a partial solution; and (v) a solution function, which will indicate when we have discovered a complete solution.

There are two ingredients that are exhibited by most problems that lend themselves to a greedy strategy:

1. Greedy Choice Property: We can make whatever choice seems best at the moment and then solve the subproblems arising after the choice is made. The choice made by a greedy algorithm may depend on choices so far. But, it cannot depend on any future choices or all the solutions to the subproblem, it progresses in a fashion making one greedy choice after another iteratively reducing each given problem into a smaller one. This is the main difference between it and dynamic programming. Dynamic programming is exhaustive and is guaranteed to find the solution. After every algorithmic stage, dynamic programming makes decisions based on the all the decisions made in the previous stage, and may reconsider the previous stage's algorithmic path to solution. A greedy algorithm makes the decision early and changes the algorithmic path after decision, and will never reconsider the old decisions. It may not be accurate for some problems.
2. Optimal Sub structure: A problem exhibits optimal sub–structure, if an optimal solution to the sub–problem contains within its optimal solution to the problem.

For most problems, greedy algorithms mostly (but not always) fail to find the globally optimal solution, because they usually do not operate exhaustively on all the data. They can make commitments to certain choices too early which prevent them from finding the best overall solution later. For example, all known greedy algorithms for the graph coloring problem and all other NP-complete problems do not consistently find optimum solutions. Nevertheless, they are useful because they are quick to think up and often give good approximations to the optimum. If a greedy algorithm can be proven to yield the global optimum for a given problem class, it typically becomes the method of choice because it is faster than other optimisation methods like dynamic programming. Examples of such greedy algorithms are *Kruskal's algorithm*, *Dijkstra's algorithms* for finding single–source shortest paths and *Prim's algorithm* for finding minimum spanning trees and the algorithm for finding optimum *Huffman trees*. The theory of *matroids* provide whole classes of such algorithms.

[141] Functions can only be memoized if they are referentially transparent, that is, if they will always return the same result given the same arguments. Operations which are not referentially transparent, but whose results are not likely to change rapidly, can still be cached with methods more complicated than memoization. In general, memoized results are not expired or invalidated later, while caches generally are. In imperative languages, both memoization and more general caching are typically implemented using some form of associative array.

In a *functional programming language* it is possible to construct a higher–order function memoize which will create a memoized function for any

Dynamic programming usually takes one of two approaches:

- Top–down approach: The problem is broken into subproblems, and these subproblems are solved and the solutions remembered, in case they need to be solved again. This is recursion and memoization combined together.
- Bottom–up approach: All subproblems that might be needed are solved in advance and then used to build up solutions to larger problems. This approach is slightly better in stack space and number of function calls, but it is sometimes not intuitive to figure out all the subproblems needed for solving given problem.

There are 3 canonical problems associated with HMMs (see, e.g., [Rab89]):

1. Given the parameters of the model, compute the probability of a particular output sequence. This problem is solved by the *forward algorithm*.
2. Given the parameters of the model, find the most likely sequence of hidden states that could have generated a given output sequence. This problem is solved by the *Viterbi algorithm*.[142]

---

referentially transparent function. In languages without higher–order functions, memoization must be implemented separately in each function that is to benefit from it.

The term 'memoization' was coined by Donald Michie in his 1968 paper 'Memo functions and machine learning' in Nature.

[142] The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states, called the *Viterbi path*, that result in a sequence of observed events, especially in the HMM context. The *forward algorithm* is a closely related algorithm for computing the probability of a sequence of observed events. These algorithms form a subset of modern information theory.

The algorithm makes a number of assumptions. First, both the observed events and hidden events must be in a sequence. This sequence often corresponds to time. Second, these two sequences need to be aligned, and an observed event needs to correspond to exactly one hidden event. Third, computing the most likely hidden sequence up to a certain point t must depend only on the observed event at point t, and the most likely sequence at point t - 1. These assumptions are all satisfied in a first-order hidden Markov model.

The terms 'Viterbi path' and 'Viterbi algorithm' are also applied to related dynamic programming algorithms that discover the single most likely explanation for an observation. For example, in stochastic parsing a dynamic programming algorithm can be used to discover the single most likely context-free derivation (parse) of a string, which is sometimes called the 'Viterbi parse'.

The Viterbi algorithm was conceived by Andrew Viterbi as an error–correction scheme for noisy digital communication links, finding universal application in decoding the convolutional codes used in both CDMA and GSM digital cellular, dial–up modems, satellite, deep–space communications, and 802.11 wireless LANs. It is now also commonly used in speech recognition, keyword spotting, computational linguistics, and bioinformatics. For example, in *speech–to–text translation*, the acoustic signal is treated as the observed sequence of events, and a string of text is considered to be the 'hidden cause' of the

3. Given an output sequence or a set of such sequences, find the most likely set of state transition and output probabilities. In other words, train the parameters of the HMM given a dataset of sequences. This problem is solved by the *Baum–Welch algorithm*.[143]

Hidden Markov models are especially known for their applications in *speech recognition*, *machine translation* and *bioinformatics*.

Bayesian Belief Networks

A Bayesian belief network is a form of probabilistic graphical model developed by Judea Pearl.[144] Bayesian network represents joint probability distribution of a set of variables with explicit independency assumptions. It is a *directed acyclic graph* with nodes representing variables and arcs representing probabilistic dependency relations among the variables.

If there is an arc from node A to another node B, then variable B depends directly on variable A and A is called a *parent node* of B. If the variable represented by a node has a known value then the node is said to be an *evidence node*. A node can represent any kind of variable, be it a measured parameter, a latent variable or a hypothesis. Nodes are not restricted to representing random variables; this is what is 'Bayesian' about a Bayesian network.

Let the variables be $X_1, \ldots, X_n$. Let $parents(A)$ be the parents of the node A. Then the joint distribution for X_1 through X_n is represented as the product of the probability distributions for $i = 1$ to n. If X_i has no parents, its probability distribution is said to be unconditional, otherwise it is conditional.

Questions about incongruent dependence among variables can be answered by studying the graph alone. It can be shown that conditional independence

acoustic signal. The Viterbi algorithm finds the most likely string of text given the acoustic signal.

[143] The *Baum–Welch algorithm* is an expectation–maximization (EM) algorithm (see [BPS70]). It can compute *maximum likelihood estimates* and *posterior–mode estimates* for the parameters (transition and emission probabilities) of an HMM, when given only emissions as training data. The algorithm has two steps: (i) Calculating the forward probability and the backward probability for each HMM state; and (ii) On the basis of this, determining the frequency of the t*ransition–emission pair* values and dividing it by the probability of the entire string. This amounts to calculating the expected count of the particular transition–emission pair. Each time a particular transition is found, the value of the quotient of the transition divided by the probability of the entire string goes up, and this value can then be made the new value of the transition.

[144] Judea Pearl is a computer scientist and statistician, best known for his prominent work on the probabilistic approach to artificial intelligence, and in particular on Bayesian belief networks. His work is also intended as a *high–level cognitive model*. He is interested in the philosophy of causality, artificial intelligence and knowledge representation, probabilistic and causal reasoning, nonstandard logics, and learning strategies. Pearl is described as 'one of the giants in the field of artificial intelligence'.

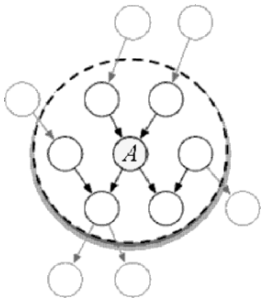

Fig. 1.6. A generic Markov blanket: the set of nodes $MB(A)$ composed of A's parents, its children, and its children's parents.

is represented in the graph by the graphical property of $d-$separation: nodes X and Y are $d-$separated in the graph, given specified evidence nodes, iff variables X and Y are independent given the corresponding evidence variables. The set of all other nodes on which node X can directly depend is given by X's *Markov blanket*.

The Markov blanket (see Figure 1.6) for a node A in a Bayesian network is the set of nodes $MB(A)$ composed of A's parents, its children, and its children's parents. In a *Markov network*, the Markov blanket of a node is its set of neighboring nodes. Every node in the network is conditionally independent of A when conditioned on the set $MB(A)$, that is, when conditioned on the Markov blanket of the node A. Formally, for distinct nodes A and B, we have

$$\Pr(A \mid MB(A), B) = \Pr(A \mid MB(A)).$$

The values of the parents and children of a node evidently give information about that node. However, its children's parents also have to be included, because they can be used to explain away the node in question. The Markov blanket of a node is interesting because it identifies all the variables that shield off the node from the rest of the network. This means that the Markov blanket of a node is the only knowledge that is needed to predict the behavior of that node.

A *causal Bayesian network* is a Bayesian network where the directed arcs of the graph are interpreted as representing *causal relations*[145] in some real domain. The directed arcs do not have to be interpreted as representing causal

[145] Recall that the philosophical *concept of causality*, the principles of causes, or causation, the working of causes, refers to the set of all particular 'causal' or 'cause–and–effect' relations. A neutral definition is notoriously hard to provide since every aspect of causation has been subject to much debate. Most generally, causation is a relationship that holds between events, properties, variables, or states of affairs. Causality always implies at least some relationship of dependency between the cause and the effect. For example, deeming something a cause may imply that, all other things being equal, if the cause occurs the effect does as well, or at least that the probability of the effect occurring increases. It is

relations; however in practice knowledge about causal relations is very often used as a guide in drawing Bayesian network graphs, thus resulting in causal Bayesian networks.

In the simplest case, a Bayesian network is specified by an expert and is then used to perform inference after some of the nodes are fixed to observed values. In some applications, such as finding *gene regulatory networks* (see [II06b]), a more complex problem of finding dependencies between variables arises. This can be solved by learning a Bayesian network that fits to the data.

Learning the structure of a Bayesian network (i.e., the graph) is a very important part of *machine learning*. Given the information that the data is being

---

also usually presumed that the cause chronologically precedes the effect. In natural languages, causal relationships can be expressed by the following causative expressions:

(i) a set of causative verbs (cause, make, create, do, effect, produce, occasion, perform, determine, influence; construct, compose, constitute; provoke, motivate, force, facilitate, induce, get, stimulate; begin, commence, initiate, institute, originate, start; prevent, keep, restrain, preclude, forbid, stop, cease);
(ii) a set of causative names (actor, agent, author, creator, designer, former, originator; antecedent, causality, causation, condition, fountain, occasion, origin, power, precedent, reason, source, spring; reason, grounds, motive, need, impulse);
(iii) a set of effective names (consequence, creation, development, effect, end, event, fruit, impact, influence, issue, outcome, outgrowth, product, result, upshot).

Causality is the centerpiece of the universe and so the main subject of human knowledge; for comprehending the nature, meaning, kinds, varieties, and ordering of cause and effect amounts to knowing the beginnings and endings of things, to uncovering the implicit mechanisms of world dynamics, or to having the fundamental scientific knowledge.

Ancient Hindu scriptures, the Upanishads (namely Chandogya Upanishad, Sarva Sara Upanishad and Mandukya Upanishad) and some other texts (namely Brahma Sutras, Yoga Vashishta, Avadhuta Gita and Astavakra Gita) mention causality. However, the mention is limited to the purpose of explaining creation of the universe: 'Cause is the effect concealed, effect is the cause revealed', which is also expressed as 'Cause is the effect unmanifested, effect is the cause manifested' (reference Complete Works of *Swami Vivekananda*, as well as *Yoga Vashishta*); 'Effect is same as cause only' (reference *Sankaracharya's* commentary on *Bhagavad Gita*).

In Metaphysics and Posterior Analytics, Aristotle stated: "All causes of things are beginnings; that we have scientific knowledge when we know the cause; that to know a thing's existence is to know the reason why it is." With this, he set the guidelines for all the subsequent causal theories by specifying the number, nature, principles, elements, varieties, order of causes as well as the modes of causation. Aristotle's account of the causes of things may be qualified as the most comprehensive model up to now.

The modern deterministic world–view is one in which the universe is nothing more than a chain of events following one after another according to the law of cause and effect.

generated by a Bayesian network and that all the variables are visible in every iteration, the following methods are used to learn the structure of the acyclic graph and the conditional probability table associated with it. The elements of a *structure–finding algorithm* are a *scoring function* and a *search strategy*. The time requirement of an exhaustive search returning back a structure that maximizes the score is superexponential in the number of variables. A local search algorithm makes incremental changes aimed at improving the score of the structure. A global search algorithm like *Markov–chain Monte–Carlo* (MCMC) can avoid getting trapped in local minima.

In order to fully specify the Bayesian network and thus fully represent the joint probability distribution, it is necessary to further specify for each node X the probability distribution for X conditional upon X's parents. The distribution of X conditional upon its parents may have any form. It is common to work with discrete or Gaussian distributions since that simplifies calculations. Sometimes only constraints on a distribution are known; one can then use the principle of maximum entropy to determine a single distribution, the one with the greatest entropy given the constraints. Analogously, in the specific context of a dynamic Bayesian network, one commonly specifies the conditional distribution for the hidden state's temporal evolution to maximize the entropy rate of the implied stochastic process. Often these conditional distributions include parameters which are unknown and must be estimated from data, sometimes using the maximum likelihood approach. Direct maximization of the likelihood (or of the posterior probability) is often complex when there are unobserved variables. A classical approach to this problem is the *expectation–maximization algorithm* which alternates computing expected values of the unobserved variables conditional on observed data, with maximizing the complete likelihood (or posterior) assuming that previously computed expected values are correct. Under mild regularity conditions this process converges on maximum likelihood (or maximum posterior) values for parameters. A more fully Bayesian approach to parameters is to treat parameters as additional unobserved variables and to compute a full posterior distribution over all nodes conditional upon observed data, then to integrate out the parameters. This approach can be expensive and lead to large dimension models, so in practise classical parameter–setting approaches are more common.

The goal of inference in Bayesian networks is typically to find the distribution of a subset of the variables, conditional upon some other subset of variables with known values (the evidence), with any remaining variables integrated out. This is known as the posterior distribution of the subset of the variables given the evidence. The posterior gives a universal sufficient statistic for detection applications, when one wants to choose values for the variable subset which minimize some expected loss function, for instance the probability of decision error.

A Bayesian network can thus be considered a mechanism for automatically constructing extensions of *Bayes' theorem* to more complex problems. Bayes' theorem relates the conditional and marginal probability distributions

of random variables. In some interpretations of probability, Bayes' theorem tells how to update or revise beliefs in light of new evidence: a posteriori. The probability of an event A conditional on another event B is generally different from the probability of B conditional on A. However, there is a definite relationship between the two, and Bayes' theorem is the statement of that relationship.

As a formal theorem, Bayes' theorem is valid in all interpretations of probability. However, frequentist and Bayesian interpretations disagree about the kinds of things to which probabilities should be assigned in applications: frequentists assigned probabilities to random events according to their frequencies of occurrence or to subsets of populations as proportions of the whole; Bayesians assign probabilities to propositions that are uncertain. A consequence is that Bayesians have more frequent occasion to use Bayes' theorem. The articles on Bayesian probability and frequentist probability discuss these debates at greater length.

Formally, *Bayes' theorem* relates the conditional and marginal probabilities of stochastic events A and B as

$$\Pr(A|B) = \frac{\Pr(B|A)\ \Pr(A)}{\Pr(B)} \propto L(A|B)\ \Pr(A),$$

where $L(A|B)$ is the likelihood of A given fixed B. Each term in Bayes' theorem has a conventional name:

$Pr(A)$ is the prior probability or marginal probability of A. It is 'prior' in the sense that it does not take into account any information about B;

$Pr(A|B)$ is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B.

$Pr(B|A)$ is the conditional probability of B given A.

$Pr(B)$ is the prior or marginal probability of B, and acts as a normalizing constant.

With this terminology, the theorem may be paraphrased as:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing constant}},$$

or, in words: *the posterior probability is proportional to the prior probability times the likelihood.* In addition, the ratio $Pr(B|A)/Pr(B)$ is sometimes called the *standardised likelihood*, so the theorem may also be paraphrased as:

$$\text{posterior} = \text{standardised likelihood} \times \text{prior}.$$

The most common exact inference methods are variable elimination which eliminates (by integration or summation) the non–observed non–query variables one by one by distributing the sum over the product, clique tree propagation which caches the computation so that the many variables can be queried

at one time, and new evidence can be propagated quickly, recursive conditioning which allows for a space-time tradeoff but still allowing for the efficiency of variable elimination when enough space is used. All of these methods have complexity that is exponential in tree width. The most common approximate inference algorithms are stochastic MCMC simulation, mini–bucket elimination which generalizes loopy belief propagation, and variational methods.

Bayesian networks are used for modelling knowledge in gene regulatory networks, medicine, engineering, text analysis, image processing, data fusion, and decision support systems.

Support Vector Machines

Recall that *support vector machines* (SVMs, see Figure 1.7) are a set of related *supervised learning* methods used for *classification* and *regression* (see [Vap95, Vap98, SS01, CS00]). Their common factor is the use of a technique known as the '*kernel trick*' to apply *linear classification* techniques to *nonlinear classification* problems.

SVMs implement the statistical learning theory. They are a radically different type of classifier from artificial neural networks (ANNs, see below) that has attracted a great deal of attention lately due to the novelty of the concepts that they bring to pattern recognition, their strong mathematical foundation, and their excellent results in practical problems. SVM represents the coupling of the following two concepts: the idea that transforming the data into a high–dimensional space makes linear discriminant functions practical, and the idea of large margin classifiers to train the standard ANNs like MLP or RBF. It is another type of a kernel classifier: it places Gaussian kernels over the data and linearly weights their outputs to create the system output. To implement the SVM–methodology, we can use the Adatron–kernel algorithm, a sophisticated nonlinear generalization of the RBF networks, which maps inputs to a high–dimensional feature space, and then optimally separates data into their respective classes by isolating those inputs, which fall close to the data boundaries. Therefore, the Adatron–kernel is especially effective in separating sets of data, which share complex boundaries, as well as for the training for nonlinearly separable patterns. The support vectors allow the network to rapidly converge on the data boundaries and consequently classify the inputs.

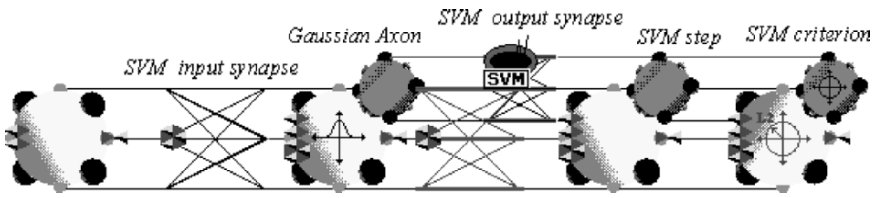

Fig. 1.7. Adatron–kernel based support vector machine (SVM) network, arranged using $NeuroSolutions^{TM}$.

The main advantage of SVMs over MLPs is that the learning task is a *convex optimization problem* which can be reliably solved even when the example data require the fitting of a very complicated function [Vap95, Vap98]. A common argument in computational learning theory suggests that it is dangerous to utilize the full flexibility of the SVM to learn the training data perfectly when these contain an amount of noise. By fitting more and more noisy data, the machine may implement a rapidly oscillating function rather than the smooth mapping which characterizes most practical learning tasks. Its prediction ability could be no better than random guessing in that case. Hence, modifications of SVM training [CS00] that allow for training errors were suggested to be necessary for realistic noisy scenarios. This has the drawback of introducing extra model parameters and spoils much of the original elegance of SVMs.

Mathematics of SVMs is based on real *Hilbert space* methods.

Linear Classification Problem

Suppose we want to classify some data points into two classes. Often we are interested in classifying data as part of a machine–learning process. These data points may not necessarily be points in $\mathbb{R}^2$ but may be multidimensional $\mathbb{R}^p$ (statistics notation) or $\mathbb{R}^n$ (computer science notation) points. We are interested in whether we can separate them by a *hyperplane*. As we examine a hyperplane, this form of classification is known as linear classification. We also want to choose a hyperplane that separates the data points 'neatly', with maximum distance to the closest data point from both classes – this distance is called the *margin*. We desire this property since if we add another data point to the points we already have, we can more accurately classify the new point since the separation between the two classes is greater. Now, if such a hyperplane exists, the hyperplane is clearly of interest and is known as the *maximum–margin hyperplane* or the *optimal hyperplane*, as are the vectors that are closest to this hyperplane, which are called the *support vectors*.

Formalization

Consider data points of the form

$$\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_n, c_n)\},$$

where the c_i is either 1 or -1; this constant denotes the class to which the point $\mathbf{x}_i$ belongs. Each $\mathbf{x}_i$ is a pD (statistics notation), or nD (computer science notation) vector of scaled $[0, 1]$ or $[-1, 1]$ values. The scaling is important to guard against variables (attributes) with larger variance that might otherwise dominate the classification. We can view this as *training data*, which denotes the correct classification which we would like the SVM to eventually distinguish, by means of the dividing hyperplane, which takes the form:

$$\mathbf{w} \cdot \mathbf{x} - b = 0.$$

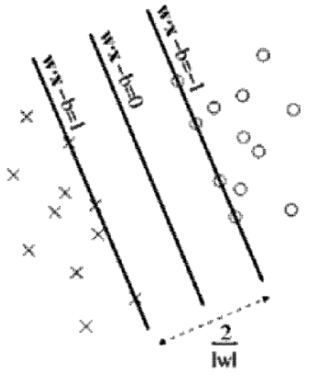

Fig. 1.8. Maximum–margin hyperplanes for a SVM trained with samples from two classes. Samples along the hyperplanes are called the support vectors.

As we are interested in the maximum margin, we are interested in the support vectors and the parallel hyperplanes (to the optimal hyperplane) closest to these support vectors in either class (see Figure 1.8). It can be shown that these parallel hyperplanes can be described by equations

$$\mathbf{w} \cdot \mathbf{x} - b = 1, \tag{1.6}$$
$$\mathbf{w} \cdot \mathbf{x} - b = -1. \tag{1.7}$$

We would like these hyperplanes to maximize the distance from the dividing hyperplane and to have no data points between them. By using geometry, we find the distance between the hyperplanes being $2/|\mathbf{w}|$, so we want to minimize $|\mathbf{w}|$. To exclude data points, we need to ensure that for all i either

$$\mathbf{w} \cdot \mathbf{x}_i - b \geq 1, \qquad \text{or}$$
$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1.$$

This can be rewritten as

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \qquad (1 \leq i \leq n). \tag{1.8}$$

The problem now is to minimize $|w|$ subject to the constraint (1.8). This is a *quadratic programming optimization* (QP) problem.

After the SVM has been trained, it can be used to classify unseen 'test' data. This is achieved using the following decision rule,

$$\hat{c} = \begin{cases} 1 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b \geq 0, \\ -1 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b \leq 0. \end{cases}$$

Writing the classification rule in its dual form reveals that classification is only a function of the support vectors, i.e., the training data that lie on the margin.

The use of the maximum–margin hyperplane is motivated by *Vapnik–Chervo-nenkis SVM theory*, which provides a probabilistic test *error bound* that is minimized when the margin is maximized. However the utility of this theoretical analysis is sometimes questioned given the large slack associated with these bounds: the bounds often predict more than 100% error rates.

The parameters of the maximum–margin hyperplane are derived by solving the optimization. There exist several specialized algorithms for quickly solving the QP problem that arises from SVMs. The most common method for solving the QP problem is Platt's *SMO algorithm*.

Nonlinear Classification

The original optimal hyperplane algorithm proposed by Vladimir Vapnik in 1963 was a *linear classifier*. However, in 1992, B. Boser, I. Guyon and Vapnik suggested a way to create nonlinear classifiers by applying the *kernel trick* (originally proposed by Aizerman) to maximum–margin hyperplanes. The resulting algorithm is formally similar, except that every *dot product* is replaced by a nonlinear *kernel* function. This allows the algorithm to fit the maximum–margin hyperplane in the transformed feature *space*. The transformation may be nonlinear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high–dimensional feature space it may be nonlinear in the original input space.

If the kernel used is a Gaussian *radial basis function*, the corresponding feature space is a *Hilbert space* of infinite dimension. Maximum margin classifiers are well *regularized*, so the infinite dimension does not spoil the results. Some common kernels include:

1. *Polynomial* (*homogeneous*):
$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d;$$
2. *Polynomial* (*inhomogeneous*):
$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + 1)^d;$$
3. *Radial Basis Function*:
$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2), \qquad \text{for} \qquad \gamma > 0;$$
4. *Gaussian radial basis function*:
$$k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}); \qquad \text{and}$$
5. *Sigmoid:*
$$k(\mathbf{x}, \mathbf{x}') = \tanh(\kappa \mathbf{x} \cdot \mathbf{x}' + c),$$
for some (not every) $\kappa > 0$ and $c < 0$.

Soft Margin

In *1995*, *Corinna Cortes* and Vapnik suggested a modified maximum margin idea that allows for mislabeled examples. If there exists no hyperplane that can split the 'yes' and 'no' examples, the so–called *soft margin method* will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. This work popularized the expression *Support Vector Machine* or *SVM*. This method introduces slack variables and the equation (1.8) now transforms to

$$c_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 - \xi_i, \qquad (1 \leq i \leq n), \tag{1.9}$$

and the optimization problem becomes

$$\min ||w||^2 + C \sum_i \xi_i \qquad \text{such that} \quad c_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 - \xi_i, \qquad (1 \leq i \leq n),$$

This constraint in (1.9) along with the objective of minimizing $|w|$ can be solved using *Lagrange multipliers* or setting up a dual optimization problem to eliminate the slack variable.

SV Regression

A version of a SVM for regression was proposed in *1995* by Vapnik, S. Golowich, and A. Smola (see [Vap98, SS01]). This method is called *support vector regression* (SVR). The model produced by support vector classification (as described above) only depends on a subset of the training data, because the *cost function* for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is close (within a threshold ε) to the model prediction.

Intelligent Agents

Recall that the *agent theory* concerns the definition of the so–called *belief–desire–intention agents* (BDI–agents, for short), as well as multi–agent systems, properties, architectures, communication, cooperation and coordination capabilities (see [RG98]).

A common definition of an agent reads: An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design requirements [Woo00].

Practical side of the agent theory concerns the agent languages and platforms for programming and experimenting with agents. According to [Fer99], a BDI–agent is a physical or virtual entity which:

1. *is capable of limited perceiving its environment* (see Figure 1.9),
2. *has only a partial representation of its environment,*

Fig. 1.9. A basic agent–environment loop (modified from [Woo00]).

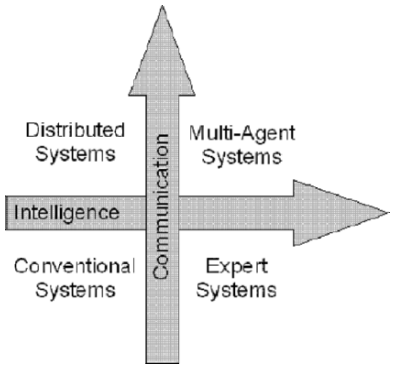

Fig. 1.10. Agent technology compared to relevant technologies.

3. *is capable of acting in an environment,*
4. *can communicate directly with other agents,*
5. *is driven by a set of tendencies,*[146]
6. *possesses resources of its own,*
7. *possesses some skills and can offer services,*
8. *may be able to reproduce itself,*
9. *whose behavior tends towards satisfying its objectives,*

– taking into account the resources and skills available to it and depending on its perception, its representation and the communications it receives. Agents' actions affect the environment which, in turn, affects future decisions of agents. The *multi–agent systems* have been successfully applied in numerous fields (see [Fer99] for the review).

Agents embody a new software development paradigm that attempts to merge some of the theories developed in artificial intelligence research with computer science. The power of agents comes from their intelligence and also their ability to communicate with each other. A simple mapping of agent technology compared to relevant technologies is illustrated in Figure 1.10. Agents can be considered as the successors of *object–oriented programming* techniques, applied to certain problem domains. However, the additional layer

[146] in the form of individual objectives or of a satisfaction/survival function which it tries to optimize

of implementation in agents provides some key functionalities and deliberately creates a separation between the implementation of an agent from the application being developed. This is done in order to achieve one of the core properties of agents, autonomy. Objects are able to assert a certain amount of control over themselves via private variables and methods, and other objects via public variables and methods. Consequently, a particular object is able to directly change public variables of other objects and also execute public methods of other objects. Hence, objects have no control over the values of public variables and who and when executes their public methods. Conversely, agents are explicitly separated, and can only request from each other to perform a particular task. Furthermore, it cannot be assumed that after a particular agent makes a request, another agent will do it. This is because performing a particular action may not be in the best interests of the other agent, in which case it would not comply [Woo00].

Types of Intelligent Agents

Here we give a general overview of different types of agents and groups them into several intuitive categories based on the method that they perform their reasoning [Woo00].

Deliberate Agents

Deliberate agents are agents that perform rational reasoning, take actions that are rational after deliberating using their *knowledge base* (KB), carefully considering the possible effects of different actions available to them. There are two subtypes of deliberate agents: *deductive reasoning agents* and *production–rule agents*.

1. Deductive reasoning agents are built using expert systems theory, they operate using an internal symbolic KB of the environment. Desired behavior is achieved by manipulating the environment and updating the KB accordingly. A utility function is implemented that provides an indication on how good a particular state is compared on what the agent should achieve. An example of the idea behind these type of agents is an agent that explores a building. It has the ability to move around and it uses a video camera, the video signal is processed and translated to some symbolic representation. As the agent explores the world it maintains a data structure of what it has explored. The internal structure of deductive reasoning agents is illustrated in Figure 1.11. There are two key problems encountered when trying to build deductive reasoning agents. Firstly, the transduction problem is the problem of translating the real world into an accurate, symbolic description in time for it to be useful. Secondly, the representation or reasoning problem is the problem of representing acquired information symbolically and getting agents to manipulate/reason with it [Woo00].

2. Production systems are also an extension of expert systems. However they place more emphasis how decisions are made based on the state of the

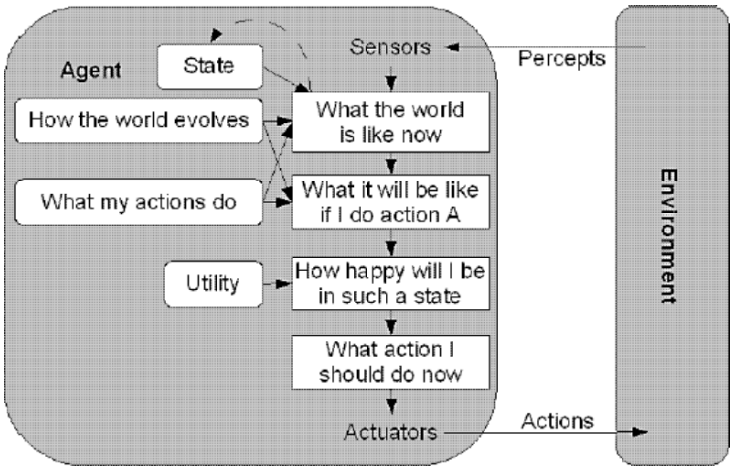

Fig. 1.11. A concept of deductive reasoning agents (modified from [RN03]).

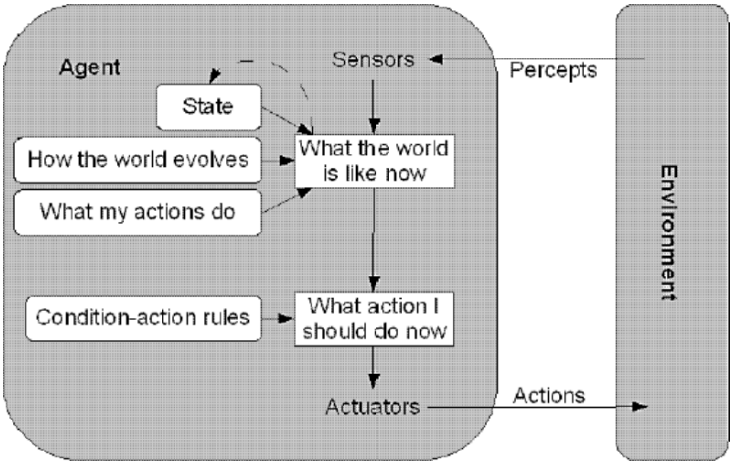

Fig. 1.12. A concept of production–rule agents (modified from [RN03]).

KB. The general structure of production system agents is illustrated in Figure 1.12. The KB is called working memory and is aimed to resemble short term memory. They also allow a designer to create a large set of condition-action rules called productions that resemble long term memory. When a production is executed it is able cause changes to the environment or directly change the working memory. This in turn possibly activates other productions. Production systems typically contain a small working memory, and a large number of rules that can be executed so fast that production systems are able to operate in real time with thousands of rules [RN03]. An example of a production–rule agent development environment is called SOAR (State,

Operator And Result). SOAR uses a KB as a problem space and production rules to look for solutions in a problem. IT has a powerful problem solving mechanism whereby every time that it is faced with more than one choice of productions (via a lack of knowledge about what is the best way to proceed) it creates an impasse that results in branching of the paths that it takes through the problem space. The impasse asserts subgoals that force the creation of sub–states of problem solving behavior with the aim to resolve the super–state impasse [Sio05].

Reactive Agents

Deliberate agents were originally developed using traditional software engineering techniques. Such techniques define pre–conditions required for operation and post–conditions that define the required output after operation. Some agents however, cannot be easily developed using this method because they maintain a constant interaction with a dynamic environment, hence they are called reactive agents. Reactive agents as especially suited for real–time applications where there are strict time constraints (i.e., milliseconds) on choosing actions.

Reactive systems are studied by behavioral means where researchers have tried to use entirely new approaches that reject any symbolic representation and decision making. Instead, they argue that intelligent and rational behavior emerges from the interaction of various simpler behaviors and is directly linked to the environment that the agent occupies [Woo00]. The general structure of reactive agents is illustrated in Figure 1.13 The main contributor of reactive agent research is Rod Brooks from MIT, with his *subsumption architecture*, where decision making is realized through a set of task–accomplishing

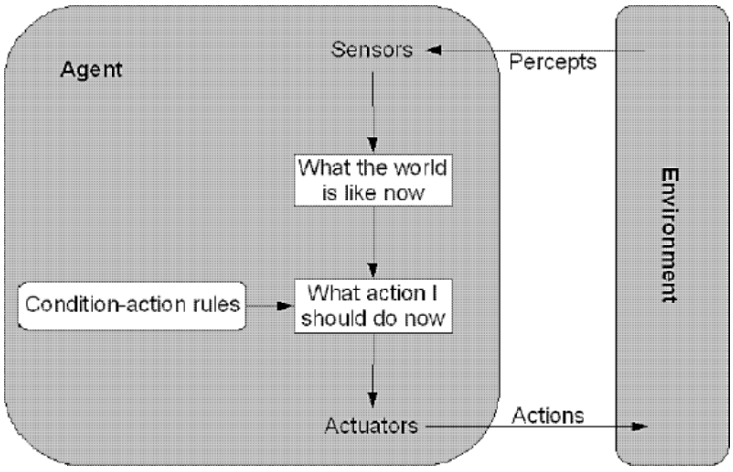

Fig. 1.13. A concept of reactive agents (modified from [RN03]).

behaviors. Behaviors are arranged into layers where lower layers have a higher priority and are able to inhibit higher layers that represent more abstract behaviors [Bro86]. A simple example of the subsumption architecture is a multi–agent system used to collect a specific type of rock scattered in a particular area on a distant planet. Agents are able to move around, collect rocks and return to the mother–ship. Due to obstacles on the surface of the planet, agents are not able to communicate directly, however they can carry special radioactive crumb that they drop on the ground for other agents to detect. The crumbs are used to leave a trail for other agents to follow. Additionally, a powerful locater signal is transmitted from the mother–ship, agents can find the ship by moving towards a stronger signal. A possible behavior architecture for this scenario are the following set of *heuristic IF–THEN rules*:

1. IF detect an obstacle THEN change direction (this rule ensures that the agent avoids obstacles when moving);
2. IF carrying samples and at the base THEN drop samples (this rule allows agent to drop samples in the mother–ship);
3. IF carrying samples and not at the base THEN drop 2 crumbs and travel up signal strength (this rule either reinforces a previous trail or creates a new one);
4. IF detect a sample THEN pick sample up (this rule collects samples);
5. IF sense crumbs THEN pick up 1 crumb and travel away from signal strength (this rule follows a crumb trail that should end at a mineral deposit; crumbs are picked up to weaken the trail such that it disappears when the mineral deposit has depleted);
6. IF true THEN move randomly (this rule explores the area until it stumbles upon a mineral deposit or a crumb trail).

Hybrid Agents

Hybrid agents are capable of expressing both reactive and pro-active behavior. They do this by breaking reactive and proactive behavior into different subsystems called layers. The lowest layer is the reactive layer and it provides immediate responses to changes for the environment, similarly to the subsumption architecture. The middle layer is the planning layer that is responsible for telling the agent what to do by reviewing internal plans, and selecting a particular plan that would be suitable for achieving a goal. The highest layer is the modelling layer that manages goals. A major issue encountered when developing solutions with hybrid reasoning agents is that agents must be able to balance the time spent between thinking and acting. This includes being able to stop planning at some point and commit to goal, even if that goal is not optimal [Woo00]. The general structure of hybrid agents is illustrated in Figure 1.14.

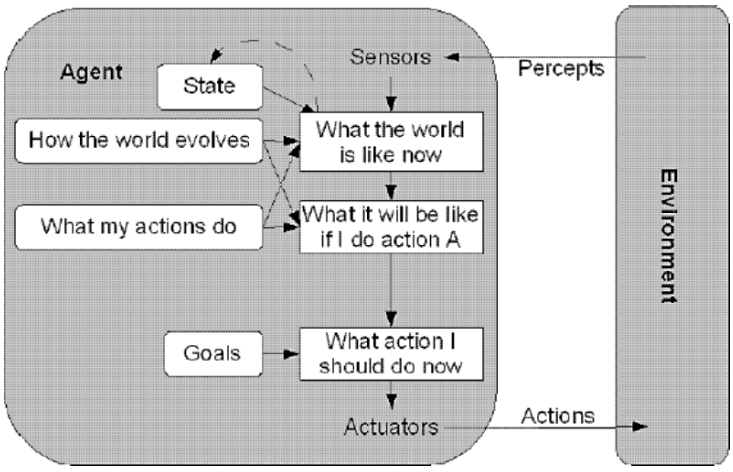

Fig. 1.14. A concept of hybrid, goal–directed agents (modified from [RN03]).

Agent–Oriented Software Development

Agent–oriented development is concerned with the techniques of software development that are specifically suited for developing agent systems. This is an important issue because existing software development techniques are unsuitable for agents as there exists a fundamental mismatch between traditional software engineering concepts and agents. Consequently, traditional techniques fail to adequately capture an agent's autonomous problem–solving behavior as well as the complex issues involved in multi–agent interaction [Sio05].

The first agent–oriented methodology was proposed by Wooldridge and is called Gaia. Gaia is deemed appropriate for agent systems with the following characteristics: (i) Agents are smart enough to require significant computational resources. (ii) Agents may be implemented using different programming languages, architectures or techniques. (iii) The system has a static organization structure such that inter–agent relationships do not change during operation. (iv) The abilities of agents and the services they provide do not change during operation. (v) The system requires only small amount of agents. Gaia splits the development process into three phases: Requirements, Analysis and Design. The requirements phase is treated in the same way as traditional systems. The analysis phase is concerned with the roles that agents play in the system as well as the interactions required between agents. The design phase is concerned with the agent types that will make up the system. The agent main services that are required to realize the agent's roles, and finally, the lines of communication between the different agents. The Gaia methodology was the inspiration for the more detailed methodology described in the next section (see [Woo00]).

Agents Environments

Agent technology has been applied to many different application areas, each focusing on a specific aspect of agents that is applicable to the domain at hand. The role that BDI–agents play in their environment distinctly depends on the application domain. The agent research community is very active and environments are mostly viewed as test–beds for developing new features in agents and showing how they are successfully used to solve a particular problem. Fortunately, in most cases this is a two–sided process, by understanding, developing and improving new agent technologies it becomes possible to solve similar real life problems. Consequently, as the underlying foundation of agent software matures, new publications describe how agents are being applied successfully in increasingly complex application domains [Sio05].

The BDI–agent is usually understood to be a *decision–maker* and anything that it interacts with, comprising everything outside the agent itself, is referred to as the *environment*. The environment has a number of features and generates *sensations* that contain some information about the features. A *situation* is commonly understood as a complete snapshot of the environment for a particular instance in time.[147] Hence, if an agent is able to get or deduce the situation of its environment it would know everything about the environment at that time. A *state* is here defined as a snapshot of the agent's beliefs corresponding to its limited understanding of the environment. This means that the state may or may not be a complete or accurate representation of the situation. This distinction supports research being conducted on improving the agent's *situation awareness* (SA), whereby SA measures how similar the state is as opposed to the situation.

The agent and the environment interact continually, the agent selects actions and the environment responds to the actions by presenting new sensations to the agent [SB98]. The interaction is normally segmented in a sequence of discrete time steps, whereby, at a particular time step the agent receives data from the environment and on that basis selects an action. In the next time step, the agent finds itself in a new state (see Figure 1.9).

Various properties of environments have been classified into six categories [RN03]:

1. *Fully observable or partially observable.* A fully observable environment provides the agent with complete, accurate and up–to–date information of the entire situation. However, as the complexity of environments increases, they become less and less observable. The physical world is considered a partially observable environment because it is not possible to know everything that happens in it [Woo00]. On the other hand, depending on the

[147] In a number of references, the term state is used with the same meaning. In this section a clear distinction is made between the two terms, a situation is defined as a complete snapshot of the real environment.

application, the environment should not be expected to be completely observable (e.g., if an agent is playing a card game it should not be expected to know the cards of every other player). Hence, in this case, even though there is hidden information in the environment and this information would be useful if the agent knew it, is not necessary for making rational decisions [SB98]. An extension of this property is when sensations received from the environment are able to summarize past sensations in a compact way such that all relevant information from the situation can be deduced. This requires that the agent maintains a history of all past sensations. When sensations succeeds in retaining all relevant information, they are said to have the Markov property. An example of a Markov sensation for a game of checkers is the current configuration of the pieces on the board, this is because it summarizes the complete sequence of sensations that led to it. Even though much of the information about the sequence is lost, all important information about the future of the game is retained. A difficulty encountered when dealing with partially observable environments is when the agent is fooled to perceiving two or more different situations as the same state, this problem is known as perceptual aliasing. If the same action is required for the different situations then aliasing is a desirable effect, and can be considered a core part of the agent's design, this technique is commonly called state generalization [SB98].

2. *Deterministic or stochastic.* Deterministic is the property when actions in the environment have a single guaranteed effect. In other words, if the same action is performed from the same situation, the result is always the same. A useful consequence of a deterministic environment is the ability to predict what will happen before an action is taken, giving rise to the possibility of evaluating multiple actions depending on their predicted effects. The physical world is classified as a stochastic environment as stated by [Woo00]. However, if an environment is partially observable it may appear to be stochastic because not all changes are observed and understood [RN03], if more detailed observations are made, including additional information, the environment becomes increasingly deterministic.
3. *Episodic or sequential.* Within an episodic environment, the situations generated are dependent on a number of distinct episodes, and there is no direct association between situations of different episodes. Episodic environments are simpler for agent development because the reasoning of the agent is based only on the current episode, there is no reason to consider future episodes [Woo00]. An important assumption made when designing agents for episodic environments, is that all episodes eventually terminate no matter what actions are selected [SB98]. This is particularly true when using learning techniques that only operate on the completion of an episode through using a captured history of situations that occurred within the episode. Actions made in sequential environments, on the other hand, affect all future decisions. Chess is an example of a sequential environment because short–term actions have long–term consequences.

4. *Static or dynamic.* A static environment is one that remains unchanged unless the agent explicitly causes changes through actions taken. A dynamic environment is one that contains other entities that cause changes in ways beyond the agents control. The physical world continuously changes with external means and is therefore considered a highly dynamic environment [Woo00]. An example of a static environment, is an agent finding its way though a 2D maze. In this case all changes are caused by the same agent. An advantage of static environments is that the agent does not need to continuously observe the environment while its deciding the next action. It can take as much time as it needs to make a decision and the environment will be the same as when previously observed [RN03].
5. *Discrete or continuous.* An environment is discrete if there is a fixed, finite number of actions and situations in it [Woo00]. Simulations and computer games are examples of discrete environments because they involve capturing actions performed by entities, processing the changes caused by the actions and providing an updated situation. Sometimes however, this process is so quick that the simulation appears to be running continuously. An example of a continuous environment is taxi driving, because the speed and location of the taxi and other cars changes smoothly over time [RN03].
6. *Single–agent or multi–agent.* Although the distinction between single and multi–agent environments may seem trivial, recent research has surfaced some interesting issues. These arise from the question of what in the environment may be viewed as another agent [RN03]. For example, does a taxi driver agent need to treat another car as an agent? What about a traffic light or a road sign? An extension to this question is when humans are included as part of the design of the system, giving rise to the new research area called human–agent teaming [Sio05].

Agents' Reasoning and Learning

The environments described above illustrate the need for *adaptation* when agent systems are required to interact with complex environments. Here we sill review how agents and humans are understood to perform reasoning and learning when they are faced with a particular environment.

Reasoning is understood as the thinking process that occurs within an agent that needs to make a particular decision. This topic has been tackled via two parallel directions with two different schools of thought. The first school of thought focuses on how agents can perform rational reasoning where the decisions made are a direct reflection of knowledge. The advantage of this approach is that decisions made by an agent can be understood simply by looking within its internal data structures, as the agent only makes decisions based on what it knows. This process includes maintaining the agent's knowledge base such that it contains accurate information about its environment, by performing operations in order to keep all knowledge consistent. Decisions

are made through a collection of rules applied on the knowledge base that define what should occur as knowledge changes [Sio05].

Another school of thought is concerned with the way that humans perform reasoning and apply any concepts developed to agent technology. Humans are known to perform practical reasoning every day, their decisions are based on their desires and their understanding in regards to how to go about achieving them. The process that takes place between observing the world, considering desires and taking actions can be broken up into four main stages, each of which consists of a number of smaller components. Through learning, it also becomes possible to create agents that are able to change the way that they were originally programmed to behave. This can be advantageous when an agent is faced with a situation that it does not know how to proceed. Furthermore, it is useful when an agent is required to improve its performance with experience.

Reasoning and Behavior

Research on artificial reasoning and behavior has been tackled from different angles that can be categorized along two main dimensions (see Figure 1.15). The vertical dimension illustrates the opposing nature of reasoning and behavior that correspond to thinking versus acting respectively. This is an important feature concept in every application using AI techniques. Great emphasis is given to the balance between processing time for making better decisions, and the required speed of operation. Approaches falling to the left side are based on how humans reason and behave while approaches falling on the right side are concerned with building systems that are rational, meaning that they are required to think and act as best they can, given their limited knowledge [RN03].

Rational Reasoning

1. Representation and search. Recall that the way that information is represented and used for intelligent problem solving forms a number of important but difficult challenges that lie within the core of AI research. Knowledge representation is concerned with the principles of correct reasoning. This involves two parallel topics of research. One side is concerned with the development of formal representation languages with the ability to maintain consistent

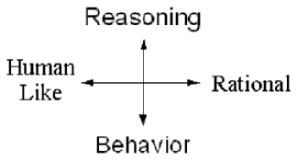

Fig. 1.15. Reasoning dimensions (modified from [RN03]).

knowledge about the world, the other side is concerned with the development of reasoning processes that bring the knowledge to life. The output of both of these areas results in a Knowledge Base (KB) system. KBs try to create a model of the real world via the collection of a number of sentences. An agent is normally able to add new sentences to the knowledge base as well as query the KB for information. Both of these tasks may require the KB to perform inference on its knowledge, where an inference is defined as the process of deriving new sentences from known information. An additional requirement of KBs is that when an agent queries the KB, the answer should be inferred from information previously added to the KB and not from unknown facts. The most important part of a KB is the logic in which the its sentences are represented. This is because all sentences in a KB are in fact expressed according to the syntax and semantics of the logic's representation language. The syntax of the logic is required for implementing well formed sentences while the semantics define the truth of each sentence with respect to a model of the environment being represented [RN03].

Problem solving using KBs involves the use of search algorithms that are able to search for solutions between different states of information within the KB. Searching involves starting from an initial state and expanding across different successor state possibilities until a solution is found. When a search algorithm is faced with a choice of possibilities to consider, each possibility is thoroughly searched before moving to the next possibility. Search however has a number of issues, including [Lug02]:
(i) Guarantee of a solution being available; (ii) Termination of the search algorithm; (iii) The optimality of a particular solution found; and (iv) The complexity of the search algorithm with respect to the time and memory usage.

State space analysis is done with the use of graphs. A graph is a set of nodes with arcs that connect them, each node can have a label to distinguish it from another node and arcs can have directions to indicate the direction of movement between the nodes. A path in the graph connects a sequence of nodes with arcs and the root is a node that has a path to all other nodes in the graph.

There are two ways to search a state space, the first way is to use data-driven search by which the search starts by a given set of facts and rules for changing states. The search proceeds until it generates a path that leads to the goal condition. Data driven search is more appropriate for problems in which the initial problem state is well defined, or there are a large number of potential goals and only a few facts to start with, or the goal state is unclear [Lug02].

The second way is to use goal-driven search by which the search starts by taking the goal state and determining what conditions must be true to move into the goal state. These conditions are then treated as subgoals to be searched. The search then continues backwards through the subgoals until it reaches the initial facts of the problem. Goal driven search is more appropriate

for problems in which the goal state is well defined, or there are a large number of initial facts making it impractical to prefer data driven search, or the initial data is not given and must be acquired by the system [Lug02].

The choice of which of the options to expand first is defined by the algorithm's search strategy. Two well known search strategies are: Breadth-first, where all successors of a given depth are expanded first before any nodes at the next level. Depth-first search involves expanding the deepest node for a particular option before moving to the next option. There are also strategies that include both elements, for example defining a depth limit for searching in a tree. It is also possible to use heuristics to help with choosing branches that are more likely to lead to an acceptable solution. Heuristics are usually applied when a problem does not have an exact solution or the computational cost to find an exact solution is too big. They reduce the state space by following the more promising paths through the state space [RN03].

An additional layer of complexity in knowledge representation and search is due to the fact that agents almost never have access a truly observable environment. Which means that agents are required to act under *uncertainty*. There are two techniques that have been used for reasoning in uncertain situations. The first involves the use of probability theory in assigning a value that represents a degree of belief in facts in the KB. The second method involves the use of *fuzzy sets* (see below) for representing how well a particular object satisfies a vague description [RN03].

2. Expert systems. Recall that knowledge–based reasoning systems are commonly called *expert systems* because they work by accumulating knowledge extracted from different sources, and use different strategies on the knowledge in order to solve problems. Simply put, expert systems try to replicate what a human expert would do if faced with the same problem. They can be classified into different categories depending on the type of problem they are used to solve [Lug02]:

- *interpretation*: making conclusions or descriptions from collections of raw data;
- *prediction/forecasting*: predicting the consequences of given situations;
- *diagnosis*: finding the cause of malfunctions based on the symptoms observed;
- *design*: finding a configuration of components that best meets performance goals when considering several design constraints;
- *planning*: finding a sequence of actions to achieve some given goals using specific starting conditions and run–time constraints;
- *monitoring*: observing a system's behavior and comparing it to its expected behavior at run–time;
- *debugging*: finding problems and repairing caused malfunctions; and
- *control*: controlling how a complex system behaves.

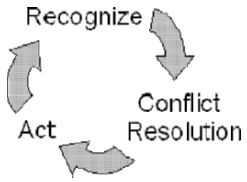

Fig. 1.16. A recognize–act operation cycle of production systems (modified from [Lug02]).

A common way to represent data in a expert systems is using first–order predicate calculus formulae. For example, the sentence 'If a bird is a crow then it is black' is represented as:

$$\forall X(crow(X) \Longrightarrow black(X)).$$

3. Production systems. They are based on a model of computation that uses search algorithms and models human problem solving. Production systems consist of production rules and a working memory. Production rules are pre-defined rules that describe a single segment of problem–solving knowledge. They are represented by a condition that determines when the production is applicable to be executed, and an action which defines what to do when executed. The working memory is an integrated KB that contains an ever–changing state of the world.

The operation of production systems generally follows a *recognize–act cycle* (see Figure 1.16). Working memory is initialized with data from the initial problem description and is subsequently updated with new information. At every step of operation, the state presented by the working memory is continuously captured as patterns and applied to conditions of productions. If a pattern is recognized against a condition, the associated production is added to a conflict set. A conflict resolution operation chooses between all enabled productions and the chosen production is fired by executing its associated action. The actions executed can have two effects. Firstly, they can cause changes to the agent's environment which indirectly changes the working memory. Secondly, they can explicitly cause changes in the working memory. The cycle then restarts using the modified working memory until a situation when no subsequent productions are enabled. Some production systems also contain the means to do backtracking when there are no further enabled productions but the goal of the system has still not been reached. Backtracking allows the system to work backwards and try some different options in order to achieve its goal [Lug02].

Human Reasoning

The so–called *practical reasoning* is concerned with studying the way that humans reason about what to do in everyday activities and applying this to

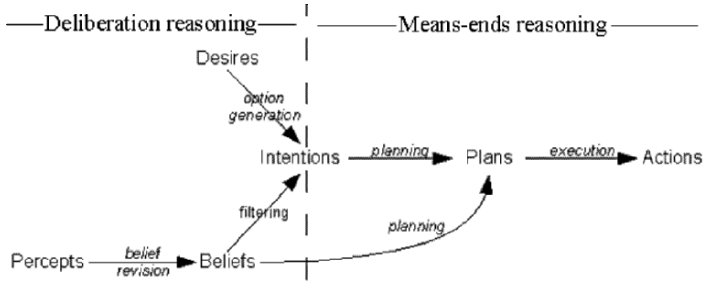

Fig. 1.17. BDI–reasoning process (modified from [Woo00]).

the design of intelligent agents. Practical reasoning is specifically geared to reasoning towards actions, it involves weighing conflicting considerations of different options that are available depending on what a person desires to do. Practical reasoning can be divided into two distinct activities (see Figure 1.17). The first activity is called *deliberation reasoning*, it involves deciding on what state to achieve. The second activity is called *means–ends reasoning* and it involves deciding on how to achieve this state of affairs [Woo00]. Recall that the central component of practical reasoning is the concept of *intention* because it is used to characterize both the action and thinking process of a person. For example 'intending to do something' characterizes a persons thinking while 'intentionally doing something' characterizes the action being taken.

The precursors of an intention are a persons's desires and beliefs and hence all of the beliefs, desires and intentions must be consistent. In other words, intending to do something must be associated with a relevant desire, as well as the belief that the intended action will help to achieve the desire. Maintaining this consistency is challenging due to the dynamic nature of desires and beliefs. Desires are always changing according to internal self-needs while beliefs are constantly updated using information obtained from senses through a process called belief revision, from the external environment.

Forming an intention involves performing two concurrent operations. Firstly, option generation uses the current desires to generate a set of possible alternatives. Secondly, filtering chooses between these alternatives based on the current intentions and beliefs. An intention also requires assigning a degree of commitment toward performing a particular action or set of actions in the future. There are four important characteristics emerging by this commitment are [Woo00]:

1. Intentions drive means-ends reasoning by forcing the agent to decide on how to achieve them.
2. Intentions persist by forcing a continuous strive to achieve them. Hence, after a particular action has failed, other alternative actions are attempted until it comes to be believed that it is not possible to achieve the intention, or the relevant desire is not longer present.

3. Intentions constrain future deliberation because it is not necessary to consider desires that are inconsistent with the current intentions.
4. Intentions influence beliefs by introducing future expectations. This is due the requirement of believing that a desired state is possible before and during execution the intention to satisfy it.

The process that occurs after forming an intention in order to take action is identified as planning, it involves selecting and advancing through a sequence of plans that dictate what actions to take. Plans are understood to consist of pre-condition that characterizes the state in which a plan is applicable for execution and a post-condition characterizes the resulting state after executing the plan. Finally, a body containing the recipe defining the actions to take [Woo00]. From the theory of practical reasoning, researchers have been able to develop intuitive agent development architectures. The transition between the theory and implementation has required the identification of equivalent software constructs for each of the BDI–components [Sio05].

Cognitive systems engineering takes into account, during the design and implementation of systems, that systems will be used by humans. It acknowledges that humans are dynamic entities that are part of the system itself but cannot be modelled as static components of a system. When humans use a system they adapt to the functional characteristics of the system. In addition, sometimes they can modify the system's functional characteristics in order to suit their own needs and preferences. This means that in order to understand the behavior of the system once the adaptation has happened is to abstract the structural elements into a purely functional level and identify and separate the functional relationships. This concept can best be understood using a simple example from [RPG94]:

> "When a novice is driving a car, it is based on an instruction manual identifying the controls of the car and explaining the use of instrument readings, that is, when to shift gears, what distance to maintain to the car ahead (depending on the speed), and how to use the steering wheel. In this way, the function of the car is controlled by discrete rules related to separate observations, and navigation depends on continuous observation of the heading error and correction by steering wheel movements. This aggregation of car characteristics and instructed input–output behavior makes it possible to drive; it initiates the novice by synchronizing them to the car functions. However, when driving skill evolves, the picture changes radically. Behavior changes from a sequence of separate acts to a complex, continuous behavioral pattern. Variables are no longer observed individually. Complex patterns of movements are synchronized with situational patterns and navigation depends on the perception of a field of safe driving. The drivers are perceiving the environment in terms of their driving goals. At this stage, the behavior of the system cannot be decomposed into

structural elements. A description must be based on abstraction into functional relationships."

A new design approach is introduced that shifts away from the traditional software engineering perspective to a functional perspective. There are two different ways to define functional characteristics. Firstly, relational representations are based on mathematical equations that relate physical, measurable environments. Secondly, casual representations are connections between different events. [RPG94] presented a framework that made it possible to relate conceptual characteristics. The framework takes into account that in order to bridge system behaviors into human profiles and preferences, several different perspectives of analysis and languages of representation are needed (see Figure 1.18).

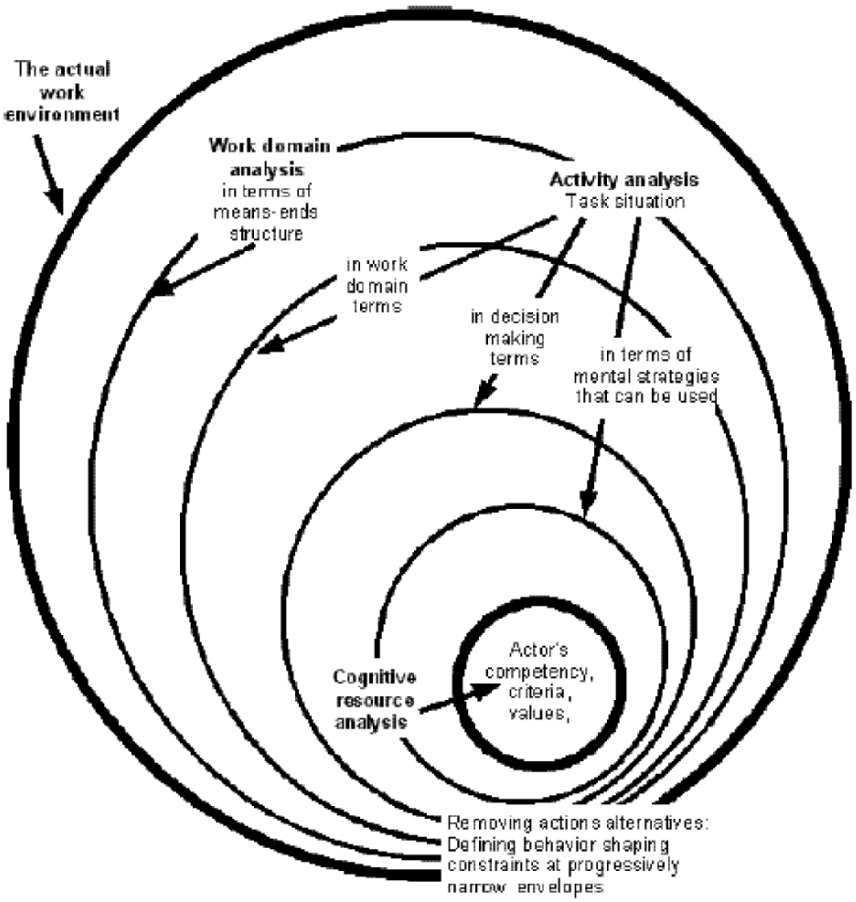

Fig. 1.18. Relating Work Environment to Cognitive Resource Profiles of Actors (adapted from [RPG94]).

In this framework, the *work domain analysis* is used to make explicit the goals, constraints and resources found in a work system. They are represented by a general inventory of system elements that are categorized by functional elements and their means-ends relations. The analysis identifies the structure and general content of the global knowledge of the work system. Activity analysis is divided into three different dimensions. Firstly, activity analysis in domain terms focuses on the freedom left for activities after the constraints posed by time and the functional space of the task. Generalizations are made in terms of objectives, functions and resources. Secondly, activity analysis in decision terms use functional languages to identify decision making functions within relevant tasks. This results of this analysis are used to identify prototype knowledge states that connect different decision functions together. Thirdly, mental strategies are used to compare task requirements with cognitive resource profiles of the individual actors and how they perform their work, thus supplies the designer with mental models, data formats and rule sets that can be incorporated into the interface of the system and used by actors of varying expertise and competence.

The *work organization analysis* is used to identify the actors involved in the decisions of different situations. This is done by finding the principles and criteria that govern the allocation of roles among the groups and group members. This allocation is dynamically dependent on circumstances and is governed by different criteria such as actor competency, access to information, minimizing communication load and sharing workload.

The *social organization analysis* focuses on the social aspect of groups working together. This is useful for understanding communication between team members, such communication may include complex information like intentions used for coordinating activities and resolving ambiguities or misinterpretations. Finally, User Analysis is used to help judge which strategy is likely to be chosen by an actor in a given situation focusing on the expertise and the performance criteria of each actor.

Rasmussen further proposes a framework for representing the various states of knowledge and information processes of human reasoning, it is called the *decision ladder* (see Figure 1.19). The ladder models the human decision making process through a set of generic operations and standardized key nodes or states of knowledge about the environment. The circles illustrated are states of knowledge and the squares are operations. The decision ladder was developed as a model for performing work domain analysis, however, the structure of the ladder is generic enough to be used as a guide in the context of describing agent reasoning.

The decision ladder can be further segmented into three levels of expertise [RPG94]. The skill (lowest) level represents very fast, automated sensory–motor performance and it is illustrated in the ladder via the heuristic shortcut links in the middle. The rule (medium) level represents the use of rules and/or procedures that have been pre–defined, or derived empirically using experience, or communicated by others, it traverses the bottom half of the ladder.

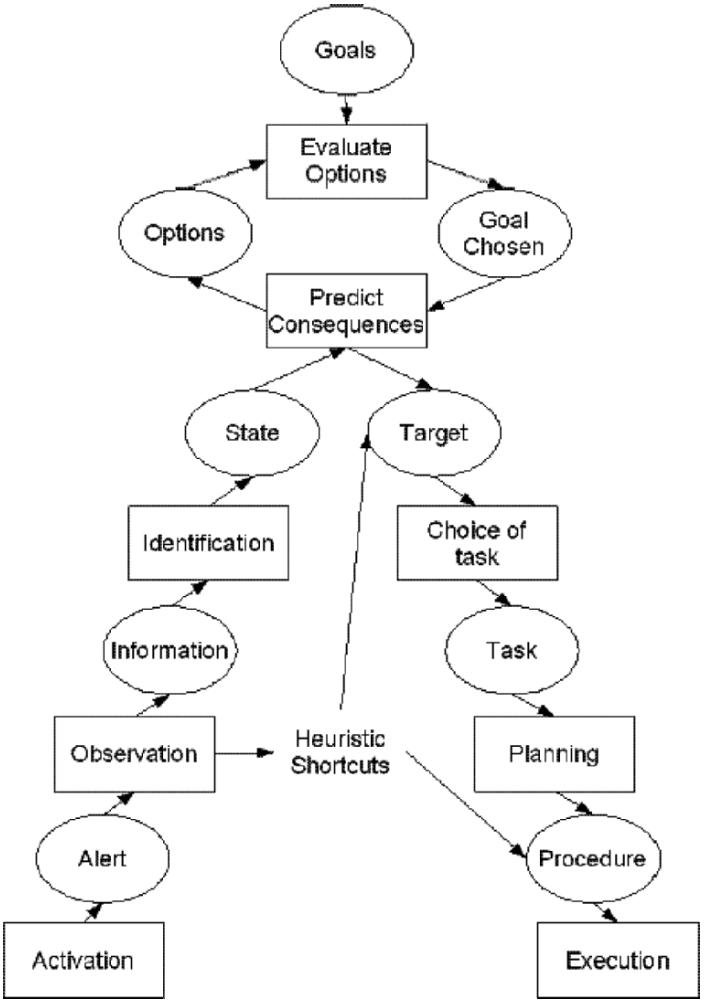

Fig. 1.19. Rassmusen's *decision ladder* (adapted from [RPG94]).

Finally, the knowledge (highest) level represents behaviors during less–familiar situations when someone is faced with an environment where there are no rules or skills available, in such cases a more detailed analysis of the environment is required with respect to the goals the agent is trying to achieve, the entire ladder is used for this case.

1.2.2 Computational Intelligence

Computational intelligence (CI) is a modern, more specifically defined AI branch. CI research aims to use learning, adaptive, or evolutionary computation to create programs that are, in some sense, intelligent. Computational

intelligence research either explicitly rejects statistical methods (as is the case with fuzzy systems), or tacitly ignores statistics (as is the case with most neural network research). In contrast, machine learning research rejects non–statistical approaches to learning, adaptivity, and optimization. Main subjects in CI, as defined by IEEE Computational Intelligence Society, are:

1. Neural networks,
2. Fuzzy systems, and
3. Evolutionary computation.

Neural Networks

Recall that an *artificial neural network* (ANN) is an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on the so–called *connectionist approach* to computation. In most cases an ANN is an *adaptive system* that changes its structure based on external or internal information that flows through the network.

In more practical terms neural networks are nonlinear statistical data modelling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

Dynamically, the ANNs are *nonlinear dynamical systems* that act as *functional approximators* [Kos92]. The ANN builds *discriminant functions* from its processing elements (PE)s. The ANN topology determines the *number* and *shape* of the discriminant functions. The shapes of the discriminant functions change with the topology, so ANNs are considered *semi–parametric classifiers*. One of the central advantages of ANNs is that they are sufficiently powerful to create arbitrary discriminant functions so ANNs can achieve optimal classification.

The placement of the discriminant functions is controlled by the network weights. Following the ideas of non–parametric training, the weights are adjusted directly from the training data without any assumptions about the data's statistical distribution. Hence one of the central issues in neural network design is to utilize systematic procedures, the so–called *training algorithm*, to modify the weights so that as accurate a classification as possible is achieved. The accuracy is quantified by an error criterion [PEL00].

The training is usually performed in the following way. First, data is presented, and an output is computed. An error is obtained by comparing the output $\{y\}$ with a desired response $\{d\}$ and it is used to modify the weights with a training algorithm. This procedure is repeated using all the data in the training set until a convergence criterion is met. Thus, in ANNs (and in adaptive systems in general) the designer does not have to specify the *parameters* of the system. They are automatically extracted from the input data and the desired response by means of the training algorithm. The two central issues in neural network design (semi–parametric classifiers) are the selection of the shape and number of the discriminant functions and their placement in pattern space such that the classification error is minimized [PEL00].

Biological Versus Artificial Neural Nets

In biological neural networks, signals are transmitted between neurons by electrical pulses (action potentials or spike trains) travelling along the axon. These pulses impinge on the afferent neuron at terminals called synapses. These are found principally on a set of branching processes emerging from the cell body (soma) known as dendrites. Each pulse occurring at a synapse initiates the release of a small amount of chemical substance or neurotransmitter which travels across the synaptic cleft and which is then received at postsynaptic receptor sites on the dendritic side of the synapse. The neurotransmitter becomes bound to molecular sites here which, in turn, initiates a change in the dendritic membrane potential. This postsynaptic potential (PSP) change may serve to increase (hyperpolarize) or decrease (depolarize) the polarization of the postsynaptic membrane. In the former case, the PSP tends to inhibit generation of pulses in the afferent neuron, while in the latter, it tends to excite the generation of pulses. The size and type of PSP produced will depend on factors such as the geometry of the synapse and the type of neurotransmitter. Each PSP will travel along its dendrite and spread over the soma, eventually reaching the base of the axon (axonhillock). The afferent neuron sums or integrates the effects of thousands of such PSPs over its dendritic tree and over time. If the integrated potential at the axonhillock exceeds a threshold, the cell fires and generates an action potential or spike which starts to travel along its axon. This then initiates the whole sequence of events again in neurons contained in the efferent pathway.

ANNs are very loosely based on these ideas. In the most general terms, a ANN consists of large numbers of simple processors linked by weighted connections. By analogy, the processing nodes may be called artificial neurons. Each node output depends only on information that is locally available at the node, either stored internally or arriving via the weighted connections. Each unit receives inputs from many other nodes and transmits its output to yet other nodes. By itself, a single processing element is not very powerful; it generates a scalar output, a single numerical value, which is a simple nonlinear function of its inputs. The power of the system emerges from the combination of many units in an appropriate way [FS92].

ANN is specialized to implement different functions by varying the connection topology and the values of the connecting weights. Complex functions can be implemented by connecting units together with appropriate weights. In fact, it has been shown that a sufficiently large network with an appropriate structure and property chosen weights can approximate with arbitrary accuracy any function satisfying certain broad constraints. In ANNs, the design motivation is what distinguishes them from other mathematical techniques: an ANN is a processing device, either an algorithm, or actual hardware, whose design was motivated by the design and functioning of animal brains and components thereof.

There are many different types of ANNs, each of which has different strengths particular to their applications. The abilities of different networks can be related to their structure, dynamics and learning methods.

Multilayer Perceptrons

The most common ANN model is the *feedforward neural network* with one input layer, one output layer, and one or more hidden layers, called *multilayer perceptron* (MLP, see Figure 1.20). This type of neural network is known as a *supervised network* because it requires a desired output in order to learn. The goal of this type of network is to *create a model* $f : x \rightarrow y$ that correctly maps the input x to the output y using historical data so that the model can then be used to produce the output when the desired output is unknown [Kos92].

In MLP the inputs are fed into the input layer and get multiplied by interconnection weights as they are passed from the input layer to the first hidden layer. Within the first hidden layer, they get summed then processed by a nonlinear function (usually the hyperbolic tangent). As the processed data leaves the first hidden layer, again it gets multiplied by interconnection weights, then summed and processed by the second hidden layer. Finally the data is multiplied by interconnection weights then processed one last time within the output layer to produce the neural network output.

MLPs are typically trained with *static backpropagation*. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than the number of network weights).

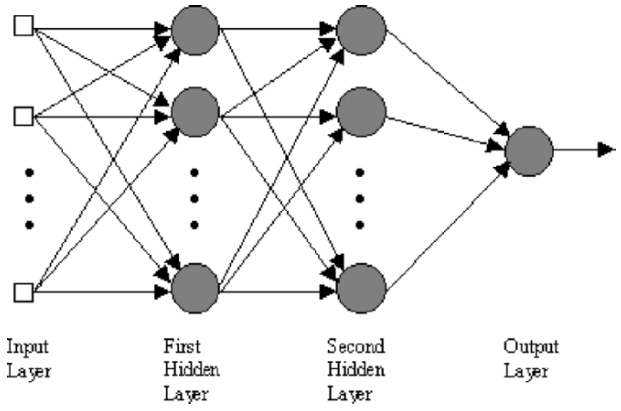

Fig. 1.20. Multilayer perceptron (MLP) with two hidden layers.

McCulloch–Pitts Processing Element

MLPs are typically composed of *McCulloch–Pitts neurons* (see [MP43]). This processing element (PE) is simply a sum–of–products followed by a threshold nonlinearity. Its input–output equation is

$$y = f(\text{net}) = f\left(w_i x^i + b\right), \qquad (i = 1, \ldots, D),$$

where D is the number of inputs, x^i are the inputs to the PE, w_i are the weights and b is a bias term (see e.g., [MP69]). The activation function is a *hard threshold* defined by *signum* function,

$$f(\text{net}) = \begin{cases} 1, & \text{for} \quad \text{net} \geq 0, \\ -1, & \text{for} \quad \text{net} < 0. \end{cases}$$

Therefore, McCulloch–Pitts PE is composed of an adaptive linear element (*Adaline*, the weighted sum of inputs), followed by a signum nonlinearity [PEL00].

Sigmoidal Nonlinearities

Besides the hard threshold defined by signum function, other nonlinearities can be utilized in conjunction with the McCulloch–Pitts PE. Let us now smooth out the threshold, yielding a sigmoid shape for the nonlinearity. The most common nonlinearities are the *logistic* and the *hyperbolic tangent threshold activation functions*,

$$\begin{aligned} \text{hyperbolic} \ : \qquad & f(\text{net}) = \tanh(\alpha\, \text{net}), \\ \text{logistic} \ : \qquad & f(\text{net}) = \frac{1}{1 + \exp(-\alpha\, \text{net})}, \end{aligned}$$

where α is a *slope parameter* and normally is set to 1. The major difference between the two sigmoidal nonlinearities is the range of their output values. The logistic function produces values in the interval $[0, 1]$, while the hyperbolic tangent produces values in the interval $[-1, 1]$. An alternate interpretation of this PE substitution is to think that the discriminant function has been generalized to

$$g(x) = f(w_i x^i + b), \qquad (i = 1, \ldots, D),$$

which is sometimes called a *ridge* function. The combination of the synapse and the tanh axon (or the sigmoid axon) is usually referred to as the modified McCulloch–Pitts PE, because they all respond to the full input space in basically the same functional form (a sum of products followed by a global nonlinearity). The output of the logistic function varies from 0 to 1. Under some conditions, the logistic function allows a very powerful interpretation of the output of the PE as a'posteriori probabilities for Gaussian–distributed input classes. The tanh is closely related to the logistic function by a linear transformation in the input and output spaces, so neural networks that use either of these can be made equivalent by changing weights and biases [PEL00].

Gradient Descent on the Net's Performance Surface

The *search* for the weights to meet a *desired response* or internal constraint is the essence of any *connectionist* computation. The central problem to be solved on the road to machine–based classifiers is how to automate the process of *minimizing the error* so that the machine can independently make these weight changes, without need for hidden agents, or external observers. The optimality criterion to be minimized is usually the *mean square error* (MSE)

$$J = \frac{1}{2N}\sum_{i=1}^{N}\varepsilon_i^2,$$

where ε_i is the instantaneous error that is added to the output y_i (the linearly fitted value), and N is the number of observations. The function $J(w)$ is called the *performance surface* (the total error surface plotted in the space of weights w).

The search for the minimum of a function can be done efficiently using a broad class of methods based on *gradient information*. The gradient has two main advantages for the search:

1. It can be computed locally, and
2. It always points in the direction of maximum change.

The *gradient of the performance surface*, $\nabla J = \nabla_w J$, is a vector (with the dimension of w) that always points toward the direction of maximum $J-$change and with a magnitude equal to the slope of the tangent of the performance surface. The minimum value of the error J_{min} depends on both the input signal x^i and the desired signal d_i,

$$J_{min} = \frac{1}{2N}\left[\sum_i d_i^2 - \frac{(d_i x^i)}{\sum_i x^i}\right], \qquad (i = 1, \ldots, D).$$

The location in coefficient space where the minimum w^* occurs also depends on both x^i and d_i. The performance surface shape depends only on the input signal x^i [PEL00].

Now, if the goal is to reach the minimum, the search must be in the direction opposite to the gradient. The overall method of gradient searching can be stated in the following way: Start the search with an arbitrary initial weight $w(0)$, where the iteration number is denoted by the index in parentheses. Then compute the gradient of the performance surface at $w(0)$, and modify the initial weight proportionally to the negative of the gradient at $w(0)$. This changes the operating point to $w(1)$. Then compute the gradient at the new position $w(1)$, and apply the same procedure again, that is,

$$w(n+1) = w(n) - \eta\nabla J(n),$$

where η is a small constant and $\nabla J(n)$ denotes the gradient of the performance surface at the nth iteration. The constant η is used to maintain stability in the search by ensuring that the operating point does not move too far along the performance surface. This search procedure is called the *steepest descent method*.

In the late 1960s, Widrow proposed an extremely elegant algorithm to estimate the gradient that revolutionized the application of gradient descent procedures. His idea is very simple: Use the instantaneous value as the estimator for the true quantity:

$$\nabla J(n) = \frac{\partial}{\partial w(n)} J \approx \frac{1}{2} \frac{\partial}{\partial w(n)} \left(\varepsilon^2(n)\right) = -\varepsilon(n)\, x(n),$$

i.e., instantaneous estimate of the gradient at iteration n is simply the product of the current input $x(n)$ to the weight $w(n)$ times the current error $\varepsilon(n)$. The amazing thing is that the gradient can be estimated with one multiplication per weight. This is the gradient estimate that led to the celebrated *least means square algorithm* (LMS):

$$w(n+1) \;=\; w(n) + \eta\varepsilon(n)\, x(n), \tag{1.10}$$

where the small constant η is called the *step size*, or the *learning rate*. The estimate will be noisy, however, since the algorithm uses the error from a single sample instead of summing the error for each point in the data set (e.g., the MSE is estimated by the error for the current sample).

Now, for fast convergence to the neighborhood of the minimum a large step size is desired. However, the solution with a large step size suffers from rattling. One attractive solution is to use a large learning rate in the beginning of training to move quickly toward the location of the optimal weights, but then the learning rate should be decreased to get good accuracy on the final weight values. This is called *learning rate scheduling*. This simple idea can be implemented with a variable step size controlled by

$$\eta(n+1) \;=\; \eta(n) - \beta,$$

where $\eta(0) = \eta_0$ is the initial step size, and β is a small constant [PEL00].

Perceptron and Its Learning Algorithm

Rosenblatt perceptron (see [Ros58b, MP69]) is a *pattern–recognition machine* that was invented in the 1950s for optical character recognition. The perceptron has an input layer fully connected to an output layer with multiple McCulloch–Pitts PEs,

$$y_i = f(\underset{i}{\text{net}}) = f(w_i x^i + b_i), \qquad (i = 1, \ldots, D),$$

where b_i is the bias for each PE. The number of outputs y_i is normally determined by the number of classes in the data. These PEs add the individual scaled contributions and respond to the entire input space.

F. Rosenblatt proposed the following procedure to directly minimize the error by changing the weights of the McCulloch–Pitts PE: Apply an input example to the network. If the output is correct do nothing. If the response is incorrect, tweak the weights and bias until the response becomes correct. Get the next example and repeat the procedure, until all the patterns are correctly classified. This procedure is called the *perceptron learning algorithm*, which can be put into the following form:

$$w(n+1) = w(n) + \eta(d(n) - y(n))\, x(n),$$

where η is the step size, y is the network output, and d is the desired response.

Clearly, the functional form is the same as in the LMS algorithm (1.10), that is, the old weights are incrementally modified proportionally to the product of the error and the input, but there is a significant difference. We cannot say that this corresponds to gradient descent since the system has a discontinuous nonlinearity. In the perceptron learning algorithm, $y(n)$ is the output of the nonlinear system. The algorithm is directly minimizing the difference between the response of the McCulloch–Pitts PE and the desired response, instead of minimizing the difference between the Adaline output and the desired response [PEL00].

This subtle modification has tremendous impact on the performance of the system. For one thing, the McCulloch–Pitts PE learns only when its output is wrong. In fact, when $y(n) = d(n)$, the weights remain the same. The net effect is that the final values of the weights are no longer equal to the linear regression result, because the nonlinearity is brought into the weight update rule. Another way of phrasing this is to say that the weight update became much more selective, effectively gated by the system performance. Notice that the LMS update is also a function of the error to a certain degree. Larger errors have more effect on the weight update than small errors, but all patterns affect the final weights implementing a 'smooth gate'. In the perceptron the net effect is that the placement of the discriminant function is no longer controlled smoothly by all the input samples as in the Adaline, only by the ones that are important for placing the discriminant function in a way that explicitly minimizes the output error.

The Delta Learning Rule

One can show that the LMS rule is equivalent to the chain rule in the computation of the *sensitivity* of the cost function J with respect to the unknowns. Interpreting the LMS equation (1.10) with respect to the sensitivity concept, we see that the gradient measures the sensitivity. LMS is therefore updating the weights proportionally to how much they affect the performance, i.e., proportionally to their sensitivity.

The LMS concept can be extended to the McCulloch–Pitts PE, which is a nonlinear system. The main question here is how can we compute the sensitivity through a nonlinearity? [PEL00] The so–called $\delta-rule$ represents a direct

extension of the LMS rule to nonlinear systems with smooth nonlinearities. In case of the McCulloch–Pitts PE, *delta–rule* reads:

$$w_i(n+1) = w_i(n) + \eta\, \varepsilon_p(n)\; x_p^i(n)\, f'(\underset{p}{\mathrm{net}}(n)),$$

where $f'(\mathrm{net})$ is the partial derivative of the static nonlinearity, such that the *chain rule* is applied to the network topology, i.e.,

$$f'(\mathrm{net})\, x^i = \frac{\partial y}{\partial w_i} = \frac{\partial y}{\partial\, \mathrm{net}} \frac{\partial}{\partial w_i}. \tag{1.11}$$

As long as the PE nonlinearity is smooth we can compute how much a change in the weight δw_i affects the output y, or from the point of view of the sensitivity, how sensitive the output y is to a change in a particular weight δw_i. Note that we compute this output sensitivity by a product of partial derivatives through intermediate points in the topology. For the nonlinear PE there is only one intermediate point, net, but we really do not care how many of these intermediate points there are. The chain rule can be applied as many times as necessary. In practice, we have an error at the output (the difference between the desired response and the actual output), and we want to adjust all the PE weights so that the error is minimized in a statistical sense. The obvious idea is to distribute the adjustments according to the sensitivity of the output to each weight.

To modify the weight, we actually *propagate back the output error* to intermediate points in the network topology and scale it along the way as prescribed by (1.11) according to the element transfer functions:

$$\begin{aligned}
\text{forward path} \;:\quad & x^i \longmapsto w_i \longmapsto \mathrm{net} \longmapsto y \\
\text{backward path 1} :\quad & w_i \overset{\partial\,\mathrm{net}\,/\partial w}{\hookleftarrow} \mathrm{net} \overset{\partial y/\partial\,\mathrm{net}}{\hookleftarrow} y \\
\text{backward path 2} :\quad & w_i \overset{\partial y/\partial w}{\hookleftarrow} y\,.
\end{aligned}$$

This methodology is very powerful, because we do not need to know explicitly the error at intermediate places, such as net. The chain rule automatically derives the error contribution for us. This observation is going to be crucial for adapting more complicated topologies and will result in the *backpropagation* algorithm, discovered in 1988 by Werbos [Wer89].

Now, several key aspects have changed in the performance surface (which describes how the cost changes with the weights) with the introduction of the nonlinearity. The nice, parabolic performance surface of the linear least squares problem is lost. The performance depends on the topology of the network through the output error, so when nonlinear processing elements are used to solve a given problem the 'performance – weights' relationship becomes nonlinear, and there is no guarantee of a single minimum. The performance surface may have several minima. The minimum that produces the smallest error in the search space is called the *global minimum*. The others are called

local minima. Alternatively, we say that the performance surface is *nonconvex*. This affects the search scheme because gradient descent uses local information to search the performance surface. In the immediate neighborhood, local minima are indistinguishable from the global minimum, so the gradient search algorithm may be caught in these suboptimal performance points, 'thinking' it has reached the global minimum [PEL00].

$\delta-$rule extended to perceptron reads:

$$w_{ij}(n+1) = w_{ij}(n) - \eta \frac{\partial J}{\partial w_{ij}} = w_{ij}(n) + \eta\, \delta_{ip}\, x_p^j\,,$$

which are local quantities available at the weight, that is, the activation x_p^j that reaches the weight w_{ij} from the input and the local error δ_{ip} propagated from the cost function J. This algorithm is local to the weight. Only the local error δ_i and the local activation x^j are needed to update a particular weight. This means that it is immaterial how many PEs the net has and how complex their interconnection is. The training algorithm can concentrate on each PE individually and work only with the local error and local activation [PEL00].

Backpropagation

The multilayer perceptron constructs input–output mappings that are a nested composition of nonlinearities, that is, they are of the form

$$y = f\left(\sum f\left(\sum (\cdot)\right)\right),$$

where the number of function compositions is given by the number of network layers. The resulting map is very flexible and powerful, but it is also hard to analyze [PEL00].

MLPs are usually trained by generalized $\delta-$rule, the so–called *backpropagation* (BP). The weight update using backpropagation is

$$w_{ij}(n+1) = w_{ij}(n) + \eta f'(\underset{i}{\text{net}}(n)) \left(\varepsilon^k(n)\, f'(\underset{k}{\text{net}}(n))\, w_{ki}(n)\right) y_j(n). \quad (1.12)$$

The summation in (1.12) is a sum of local errors δ_k at each network output PE, scaled by the weights connecting the output PEs to the ith PE. Thus the term in parenthesis in (1.12) effectively computes the total error reaching the ith PE from the output layer (which can be thought of as the ith PE's contribution to the output error). When we pass it through the ith PE nonlinearity, we have its local error, which can be written as

$$\delta_i(n) = f'(\underset{i}{\text{net}}(n))\, \delta^k w_{ki}(n).$$

Thus there is a unifying link in all the gradient–descent algorithms. All the weights in gradient descent learning are updated by multiplying the local error

$\delta_i(n)$ by the local activation $x^j(n)$ according to Widrow's estimation of the instantaneous gradient first shown in the LMS rule:

$$\Delta w_{ij}(n) = \eta\, \delta_i(n)\, y_j(n).$$

What differs is the calculation of the local error, depending on whether the PE is linear or nonlinear and if the weight is attached to an output PE or a hidden–layer PE [PEL00].

Momentum Learning

Momentum learning is an improvement to the straight gradient–descent search in the sense that a memory term (the past increment to the weight) is used to speed up and stabilize convergence. In *momentum learning* the equation to update the weights becomes

$$w_{ij}(n+1) = w_{ij}(n) + \eta\, \delta_i(n)\, x_j(n) + \alpha\, (w_{ij}(n) - w_{ij}(n-1))\,,$$

where α is the momentum constant, usually set between 0.5 and 0.9. This is called momentum learning due to the form of the last term, which resembles the momentum in mechanics. Note that the weights are changed proportionally to how much they were updated in the last iteration. Thus if the search is going down the hill and finds a flat region, the weights are still changed, not because of the gradient (which is practically zero in a flat spot), but because of the rate of change in the weights. Likewise, in a narrow valley, where the gradient tends to bounce back and forth between hillsides, the momentum stabilizes the search because it tends to make the weights follow a smoother path. Imagine a ball (weight vector position) rolling down a hill (performance surface). If the ball reaches a small flat part of the hill, it will continue past this local minimum because of its momentum. A ball without momentum, however, will get stuck in this valley. Momentum learning is a robust method to speed up learning, and is usually recommended as the default search rule for networks with nonlinearities.

Advanced Search Methods

The popularity of *gradient descent method* is based more on its simplicity (it can be computed locally with two multiplications and one addition per weight) than on its search power. There are many other search procedures more powerful than backpropagation. For example, *Newtonian method* is a second–order method because it uses the information on the curvature to adapt the weights. However Newtonian method is computationally much more costly to implement and requires information not available at the PE, so it has been used little in neurocomputing. Although more powerful, Newtonian method is still a local search method and so may be caught in local minima or diverge due to the difficult neural network performance landscapes. Other techniques such

as *simulated annealing*[148] and *genetic algorithms* (GA)[149] are global search procedures, that is, they can avoid local minima. The issue is that they are more costly to implement in a distributed system like a neural network, either because they are inherently slow or because they require nonlocal quantities [PEL00].

The problem of search with local information can be formulated as an approximation to the functional form of the *matrix cost function* $J(\mathbf{w})$ at the operating point $\mathbf{w}_0$. This immediately points to the Taylor series expansion of J around $\mathbf{w}_0$,

$$J(\mathbf{w}-\mathbf{w}_0) = J_0 + (\mathbf{w}-\mathbf{w}_0)\nabla J_0 + \frac{1}{2}(\mathbf{w}-\mathbf{w}_0)\mathbf{H}_0(\mathbf{w}-\mathbf{w}_0)^T + \cdots ,$$

where ∇J is our familiar gradient, and $\mathbf{H}$ is the Hessian matrix, that is, the matrix of second derivatives with entries

$$H_{ij}(\mathbf{w}_0) = \left.\frac{\partial^2 J(w)}{\partial w_i \partial w_j}\right|_{w=w_0} ,$$

evaluated at the operating point. We can immediately see that the Hessian cannot be computed with the information available at a given PE, since it uses information from two different weights. If we differentiate J with respect to the weights, we get

$$\nabla J(\mathbf{w}) = \nabla J_0 + \mathbf{H}_0(\mathbf{w}-\mathbf{w}_0) + \cdots \tag{1.13}$$

so we can see that to compute the full gradient at $\mathbf{w}$ we need all the higher terms of the derivatives of J. This is impossible. Since the performance surface tends to be bowl shaped (quadratic) near the minimum, we are normally interested only in the first and second terms of the expansion [PEL00].

If the expansion of (1.13) is restricted to the first term, we get the gradient–search methods (hence they are called *first–order–search methods*), where the gradient is estimated with its value at $\mathbf{w}_0$. If we expand to use the second–order term, we get *Newton method* (hence the name second–order method). If we equate the truncated relation (1.13) to 0 we immediately get

$$w = w_0 - \mathbf{H}_0^{-1}\nabla J_0 ,$$

[148] Simulated annealing is a global search criterion by which the space is searched with a random rule. In the beginning the variance of the random jumps is very large. Every so often the variance is decreased, and a more local search is undertaken. It has been shown that if the decrease of the variance is set appropriately, the global optimum can be found with probability one. The method is called simulated annealing because it is similar to the annealing process of creating crystals from a hot liquid.

[149] Genetic algorithms are global search procedures proposed by J. Holland that search the performance surface, concentrating on the areas that provide better solutions. They use 'generations' of search points computed from the previous search points using the operators of crossover and mutation (hence the name).

which is the equation for the Newton method, which has the nice property of quadratic termination (it is guaranteed to find the exact minimum in a finite number of steps for quadratic performance surfaces). For most quadratic performance surfaces it can converge in one iteration.

The real difficulty is the memory and the computational cost (and precision) to estimate the Hessian. Neural networks can have thousands of weights, which means that the Hessian will have millions of entries. This is why methods of approximating the Hessian have been extensively researched. There are two basic classes of approximations [PEL00]:

1. Line search methods, and
2. Pseudo–Newton methods.

The information in the first type is restricted to the gradient, together with line searches along certain directions, while the second seeks approximations to the Hessian matrix. Among the line search methods probably the most effective is the *conjugate gradient method*. For quadratic performance surfaces the conjugate gradient algorithm preserves quadratic termination and can reach the minimum in D steps, where D is the dimension of the weight space. Among the Pseudo–Newton methods probably the most effective is the *Levenberg–Marquardt algorithm* (LM), which uses the Gauss–Newton method to approximate the Hessian. LM is the most interesting for neural networks, since it is formulated as a sum of quadratic terms just like the cost functions in neural networks.

The *extended Kalman filter* (EKF) forms the basis of a second–order neural network training method that is a practical and effective alternative to the batch–oriented, second–order methods mentioned above. The essence of the recursive EKF procedure is that, during training, in addition to evolving the weights of a network architecture in a sequential (as opposed to batch) fashion, an approximate error covariance matrix that encodes second–order information about the training problem is also maintained and evolved.

Homotopy Methods

The most popular method for solving nonlinear equations in general is the *Newton–Raphson method*. Unfortunately, this method sometimes fails, especially in cases when nonlinear equations possess multiple solutions (zeros). An emerging family of methods that can be used in such cases are homotopy (continuation) methods. These methods are robust and have good convergence properties.

Homotopy methods or *continuation methods* have increasingly been used for solving variety of nonlinear problems in fluid dynamics, structural mechanics, systems identifications, and integrated circuits (see [Wat90]). These methods, popular in mathematical programming, are globally convergent

provided that certain coercivity and continuity conditions are satisfied by the equations that need to be solved [Wat90]. Moreover, they often yield all the solutions to the nonlinear system of equations.

The idea behind a homotopy or continuation method is to embed a parameter λ in the nonlinear equations to be solved. This is why they are sometimes referred to as *embedding methods*. Initially, parameter λ is set to zero, in which case the problem is reduced to an easy problem with a known or easily–found solution. The set of equations is then gradually deformed into the originally posed difficult problem by varying the parameter λ . The original problem is obtained for $\lambda = 1$. Homotopies are a class of continuation methods, in which parameter λ is a function of a path arc length and may actually increase or decrease as the path is traversed. Provided that certain coercivity conditions imposed on the nonlinear function to be solved are satisfied, the homotopy path does not branch (bifurcate) and passes through all the solutions of the nonlinear equations to be solved.

The zero curve of the homotopy map can be tracked by various techniques: an *ODE–algorithm*, a *normal flow algorithm*, and an *augmented Jacobian matrix algorithm*, among others [Wat90].

As a typical example, homotopy techniques can be applied to find the zeros of the gradient function $F : \mathbb{R}^N \to \mathbb{R}^N$, such that

$$F(\theta) = \frac{\partial E(\theta)}{\partial \theta_k}, \qquad 1 \leq k \leq N,$$

where $E = (\theta)$ is the certain error function dependent on N parameters θ_k. In other words, we need to solve a system of nonlinear equations

$$F(\theta) = 0. \tag{1.14}$$

In order to solve equation (1.14), we can create a linear homotopy function

$$H(\theta, \lambda) = (1 - \lambda)(\theta - a) + \lambda F(\theta),$$

where a is an arbitrary starting point. Function $H(\theta, \lambda)$ has properties that equation $H(\theta, 0) = 0$ is easy to solve, and that $H(\theta, 1) \equiv F(\theta)$.

ANNs as Functional Approximators

The *universal approximation theorem* of Kolmogorov states [Hay94]:
Let $\phi(\cdot)$ be a nonconstant, bounded, and monotone–increasing continuous (C^0) function. Let I^N denote ND unit hypercube $[0, 1]^N$. The space of C^0-functions on I^N is denoted by $C(I^N)$. Then, given any function $f \in C(I^N)$ and $\epsilon > 0$, there exist an integer M and sets of real constants $\alpha_i, \theta_i, \omega_{ij}$, $i = 1, \ldots, M$; $j = 1, \ldots, N$ such that we may define

$$F(x_1, \ldots, x_N) \;=\; \alpha_i \phi(\omega_{ij} x_j - \theta_i),$$

as an approximate realization of the function $f(\cdot)$; that is

$$|F(x_1, \ldots, x_N) \;-\; f(x_1, \ldots, x_N)| \;<\; \epsilon \qquad \text{for all} \quad \{x_1, \ldots, x_N\} \in I^N.$$

This theorem is directly applicable to *multilayer perceptrons*. First, the logistic function $1/[1+\exp(-v)]$ used as the sigmoidal nonlinearity in a neuron model for the construction of a multilayer perceptron is indeed a nonconstant, bounded, and monotone–increasing function; it therefore satisfies the conditions imposed on the function $\phi(\cdot)$. Second, the upper equation represents the output of a multilayer perceptron described as follows:

1. The network has n input nodes and a single hidden layer consisting of M neurons; the inputs are denoted by $x_1, \ldots, x_N$.
2. ith hidden neuron has synaptic weights $\omega_{i1}, \ldots, \omega_{iN}$ and threshold θ_i.
3. The network output y_j is a linear combination of the outputs of the hidden neurons, with $\alpha_i, \ldots, \alpha_M$ defining the coefficients of this combination.

The theorem actually states that a single hidden layer is sufficient for a multilayer perceptron to compute a uniform ϵ approximation to a given training set represented by the set of inputs $x_1, \ldots, x_N$ and desired (target) output $f(x_1, \ldots, x_N)$. However, the theorem does not say that a single layer is *optimum* in the sense of learning time or ease of implementation.

Recall that training of multilayer perceptrons is usually performed using a certain clone of the BP algorithm (1.2.2). In this forward–pass/backward–pass gradient–descending algorithm, the adjusting of synaptic weights is defined by the extended $\delta-$*rule*, given by equation

$$\Delta\omega_{ji}(N) = \eta \cdot \delta_j(N) \cdot y_i(N), \tag{1.15}$$

where $\Delta\omega_{ji}(N)$ corresponds to the *weight correction*, η is the *learning–rate parameter*, $\delta_j(N)$ denotes the *local gradient* and $y_i(N)$ – the *input signal of neuron* j; while the *cost function* E is defined as the instantaneous sum of squared errors e_j^2

$$E(n) = \frac{1}{2}\sum_j e_j^2(N) = \frac{1}{2}\sum_j [d_j(N) - y_j(N)]^2, \tag{1.16}$$

where $y_j(N)$ is the output of jth neuron, and $d_j(N)$ is the desired (target) response for that neuron. The slow BP convergence rate (1.15–1.16) can be accelerated using the faster LM algorithm (see subsection 1.2.2 above), while its robustness can be achieved using an appropriate fuzzy controller (see subsection (1.2.2) below).

Summary of Supervised Learning Methods

Gradient Descent Method

Given the $(D+1)$D weights vector $\mathbf{w}(n) = [w_0(n), \ldots, w_D(n)]^T$ (with $w_0 = bias$), and the correspondent MSE–gradient (including partials of MSE w.r.t. weights)

$$\nabla \mathbf{e} = \left[\frac{\partial e}{\partial w_0}, \ldots, \frac{\partial e}{\partial w_D} \right]^T,$$

and the learning rate (step size) η, we have the vector learning equation

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta \nabla \mathbf{e}(n),$$

which in index form reads

$$w_i(n+1) = w_i(n) - \eta \nabla e_i(n).$$

LMS Algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta \varepsilon(n)\, x(n),$$

where x is an input (measurement) vector, and ε is a zero–mean Gaussian noise vector uncorrelated with input, or

$$w_i(n+1) = w_i(n) + \eta \varepsilon(n)\, x^i(n).$$

Newton's Method

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta\, \mathbf{R}^{-1} \mathbf{e}(n),$$

where $\mathbf{R}$ is input (auto)correlation matrix, or

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta\, \mathbf{R}^{-1} \varepsilon(n)\, x(n),$$

Conjugate Gradient Method

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \eta\, \mathbf{p}(n), \\ \mathbf{p}(n) &= -\nabla \mathbf{e}(n) + \beta(n) \mathbf{p}(n-1), \\ \beta(n) &= \frac{\nabla \mathbf{e}(n)^T \nabla \mathbf{e}(n)}{\nabla \mathbf{e}(n-1)^T \nabla \mathbf{e}(n-1)}. \end{aligned}$$

Levenberg–Marquardt Algorithm

Putting

$$\nabla \mathbf{e} = \mathbf{J}^T \mathbf{e},$$

where $\mathbf{J}$ is the Jacobian matrix, which contains first derivatives of the network errors with respect to the weights and biases, and $\mathbf{e}$ is a vector of network errors, LM algorithm reads

$$\mathbf{w}(n+1) = \mathbf{w}(n) - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e}. \tag{1.17}$$

Generalized Feedforward Nets

The *generalized feedforward network* (GFN, see Figure 1.21) is a generalization of MLP, such that connections can jump over one or more layers, which in practice, often solves the problem much more efficiently than standard MLPs. A classic example of this is the two–spiral problem, for which standard MLP requires hundreds of times more training epochs than the generalized feedforward network containing the same number of processing elements. Both MLPs and GFNs are usually trained using a variety of backpropagation techniques and their enhancements like the nonlinear LM algorithm (1.17). During training in the spatial processing, the weights of the GFN converge iteratively to the analytical solution of the 2D Laplace equation.

Modular Feedforward Nets

The *modular feedforward networks* are a special class of MLP. These networks process their input using several parallel MLPs, and then recombine the results. This tends to create some structure within the topology, which

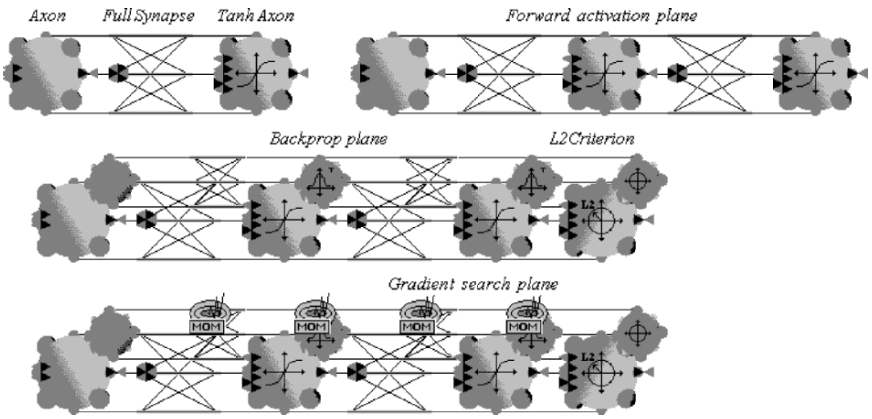

Fig. 1.21. Generalized feedforward network (GFN), arranged using *NeuroSolutions*^{TM}.

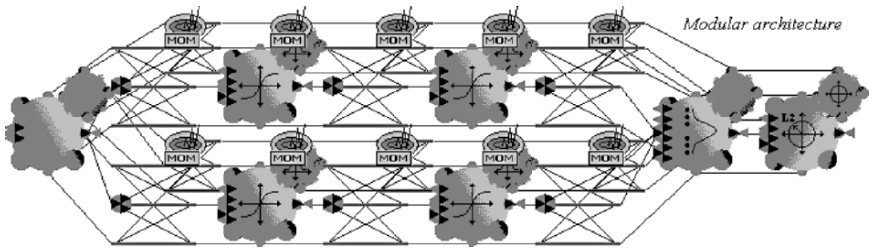

Fig. 1.22. Modular feedforward network, arranged using $NeuroSolutions^{TM}$.

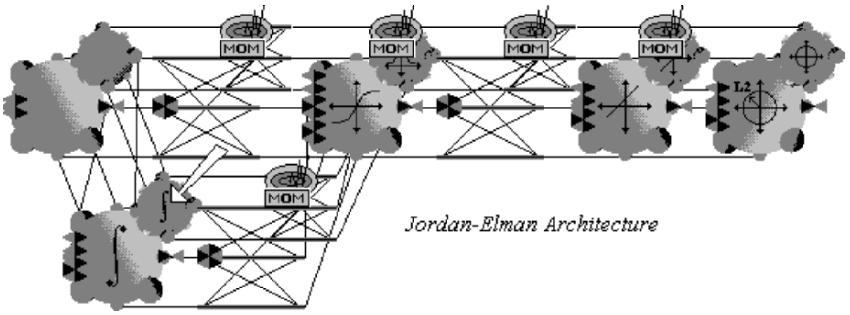

Fig. 1.23. Jordan and Elman network, arranged using $NeuroSolutions^{TM}$.

will foster specialization of function in each submodule (see Figure 1.22). In contrast to the MLP, modular networks do not have full inter–connectivity between their layers. Therefore, a smaller number of weights are required for the same size network (i.e., the same number of PEs). This tends to speed up training times and reduce the number of required training exemplars. There are many ways to segment a MLP into modules. It is unclear how to best design the modular topology based on the data. There are no guarantees that each module is specializing its training on a unique portion of the data.

Jordan and Elman Nets

Jordan and Elman networks (see [Elm90]) extend the multilayer perceptron with context units, which are processing elements (PEs) that remember past activity. Context units provide the network with the ability to extract temporal information from the data. In the Elman network, the activity of the first hidden PEs are copied to the context units, while the Jordan network copies the output of the network (see Figure 1.23). Networks which feed the input and the last hidden layer to the context units are also available.

Kohonen Self–Organizing Map

Kohonen self–organizing map (SOM, see Figure 1.24) is widely used for image pre–processing as well as a pre–processing unit for various hybrid architectures. SOM is a winner–take–all neural architecture that quantizes the input

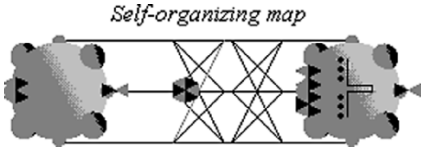

Fig. 1.24. Kohonen self–organizing map (SOM) network, arranged using $NeuroSolutions^{TM}$.

space, using a distance metric, into a discrete feature output space, where neighboring regions in the input space are neighbors in the discrete output space. SOM is usually applied to neighborhood clustering of random points along a circle using a variety of distance metrics: Euclidean, L^1, L^2, and L^n, Machalanobis, etc. The basic SOM architecture consists of a layer of Kohonen synapses of three basic forms: line, diamond and box, followed by a layer of winner–take–all axons. It usually uses added Gaussian and uniform noise, with control of both the mean and variance. Also, SOM usually requires choosing the proper initial neighborhood width as well as annealing of the neighborhood width during training to ensure that the map globally represents the input space.

The Kohonen SOM algorithm is defined as follows: Every stimulus $\mathbf{v}$ of an Euclidian input space V is mapped to the neuron with the position $\mathbf{s}$ in the neural layer R with the highest neural activity, the 'center of excitation' or 'winner', given by the condition

$$|\mathbf{w_s} - \mathbf{v}| = \min_{\mathbf{r}\in R} |\mathbf{w_r} - \mathbf{v}|,$$

where $|.|$ denotes the Euclidian distance in input space. In the Kohonen model the learning rule for each synaptic weight vector $\mathbf{w_r}$ is given by

$$\mathbf{w}_\mathbf{r}^{\mathsf{new}} = \mathbf{w}_\mathbf{r}^{\mathsf{old}} + \eta \cdot g_{\mathbf{rs}} \cdot (\mathbf{v} - \mathbf{w}_\mathbf{r}^{\mathsf{old}}), \tag{1.18}$$

with $g_{\mathbf{rs}}$ as a gaussian function of Euclidian distance $|\mathbf{r} - \mathbf{s}|$ in the neural layer. Topology preservation is enforced by the common update of all weight vectors whose neuron $\mathbf{r}$ is adjacent to the center of excitation $\mathbf{s}$. The function $g_{\mathbf{rs}}$ describes the topology in the neural layer. The parameter η determines the speed of learning and can be adjusted during the learning process.

Radial Basis Function Nets

The *radial basis function network* (RBF, see Figure 1.25) provides a powerful alternative to MLP for function approximation or classification. It differs from MLP in that the overall input–output map is constructed from local contributions of a layer of Gaussian axons. It trains faster and requires fewer training samples than MLP, using the hybrid supervised/unsupervised method. The unsupervised part of an RBF network consists of a competitive synapse followed by a layer of Gaussian axons. The means of the Gaussian axons are

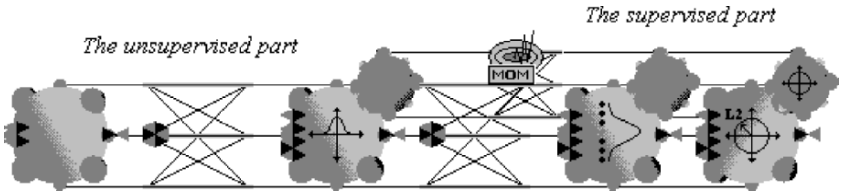

Fig. 1.25. Radial basis function network, arranged using $NeuroSolutions^{TM}$.

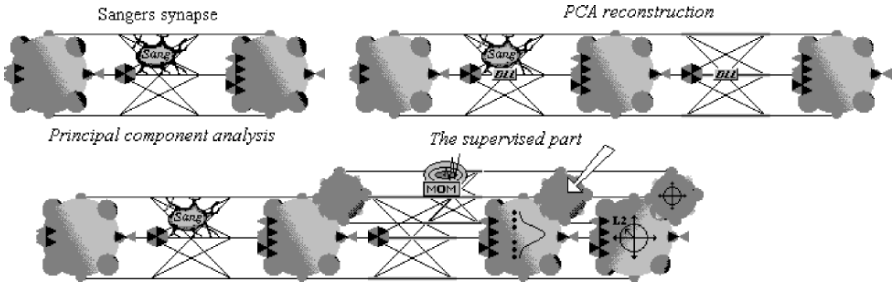

Fig. 1.26. Principal component analysis (PCA) network, arranged using $NeuroSolutions^{TM}$.

found through competitive clustering and are, in fact, the weights of the Conscience synapse. Once the means converge the variances are calculated based on the separation of the means and are associated with the Gaussian layer. Having trained the unsupervised part, we now add the supervised part, which consists of a single–layer MLP with a soft–max output.

Principal Component Analysis Nets

The *principal component analysis networks* (PCAs, see Figure 1.26) combine unsupervised and supervised learning in the same topology. Principal component analysis is an unsupervised linear procedure that finds a set of uncorrelated features, principal components, from the input. A MLP is supervised to perform the nonlinear classification from these components. More sophisticated are the *independent component analysis networks* (ICAs).

Co–active Neuro–Fuzzy Inference Systems

The *co–active neuro–fuzzy inference system* (CANFIS, see Figure 1.27), which integrates adaptable fuzzy inputs with a modular neural network to rapidly and accurately approximate complex functions. Fuzzy–logic inference systems (see next section) are also valuable as they combine the explanatory nature of rules (membership functions) with the power of 'black box' neural networks.

Genetic ANN–Optimization

Genetic optimization, added to ensure and speed–up the convergence of all other ANN–components, is a powerful tool for enhancing the efficiency and

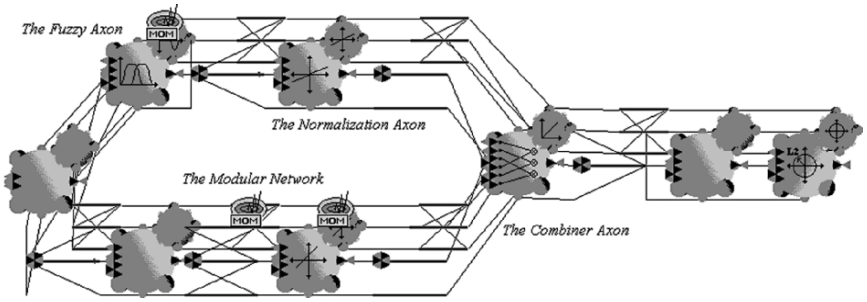

Fig. 1.27. Co–active neuro–fuzzy inference system (CANFIS) network, arranged using $NeuroSolutions^{TM}$.

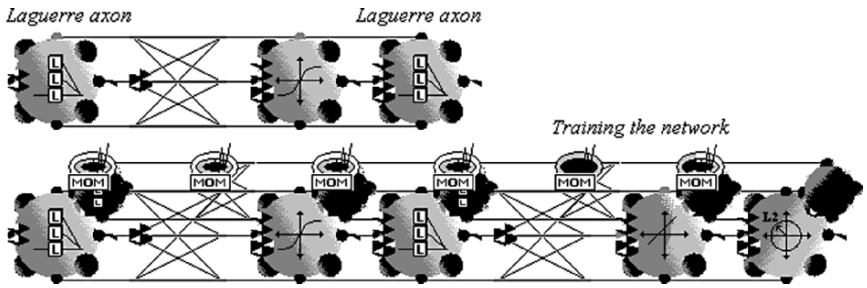

Fig. 1.28. Time–lagged recurrent network (TLRN), arranged using $NeuroSolutions^{TM}$.

effectiveness of a neural network. Genetic optimization can fine–tune network parameters so that network performance is greatly enhanced. Genetic control applies a *genetic algorithm* (GA, see next section), a part of broader *evolutionary computation*, see MIT journal with the same name) to any network parameters that are specified. Also through the *genetic control*, GA parameters such as mutation probability, crossover type and probability, and selection type can be modified.

Time–Lagged Recurrent Nets

The *time–lagged recurrent networks* (TLRNs, see Figure 1.28) are MLPs extended with short term memory structures [Wer90]. Most real–world data contains information in its time structure, i.e., how the data changes with time. Yet, most neural networks are purely static classifiers. TLRNs are the state of the art in nonlinear time series prediction, system identification and temporal pattern classification. Time–lagged recurrent nets usually use memory Axons, consisting of IIR filters with local adaptable feedback that act as a variable memory depth. The time–delay neural network (TDNN) can be considered a special case of these networks, examples of which include the Gamma and Laguerre structures. The Laguerre axon uses locally recurrent

all–pass IIR filters to store the recent past. They have a single adaptable parameter that controls the memory depth. Notice that in addition to providing memory for the input, we have also used a Laguerre axon after the hidden Tanh axon. This further increases the overall memory depth by providing memory for that layer's recent activations.

Fully Recurrent ANNs

The *fully recurrent networks* feed back the hidden layer to itself. Partially recurrent networks start with a fully recurrent net and add a feedforward connection that bypasses the recurrency, effectively treating the recurrent part as a state memory. These recurrent networks can have an infinite memory depth and thus find relationships through time as well as through the instantaneous input space. Most real–world data contains information in its time structure. Recurrent networks are the state of the art in nonlinear time series prediction, system identification, and temporal pattern classification. In case of large number of neurons, here the firing states of the neurons or their membrane potentials are the microscopic stochastic dynamical variables, and one is mostly interested in quantities such as average state correlations and global information processing quality, which are indeed measured by macroscopic observables. In contrast to layered networks, one cannot simply write down the values of successive neuron states for models of recurrent ANNs; here they must be solved from (mostly stochastic) coupled dynamic equations. For nonsymmetric networks, where the asymptotic (stationary) statistics are not known, dynamical techniques from non–equilibrium statistical mechanics are the only tools available for analysis. The natural set of macroscopic quantities (or order parameters) to be calculated can be defined in practice as the smallest set which will obey closed deterministic equations in the limit of an infinitely large network.

Being high–dimensional nonlinear systems with extensive feedback, the dynamics of recurrent ANNs are generally dominated by a wealth of attractors (fixed–point attractors, limit–cycles, or even more exotic types), and the practical use of recurrent ANNs (in both biology and engineering) lies in the potential for creation and manipulation of these attractors through adaptation of the network parameters (synapses and thresholds) (see [Hop82, Hop84]). Input fed into a recurrent ANN usually serves to induce a specific initial configuration (or firing pattern) of the neurons, which serves as a cue, and the output is given by the (static or dynamic) attractor which has been triggered by this cue. The most familiar types of recurrent ANN models, where the idea of creating and manipulating attractors has been worked out and applied explicitly, are the so–called *attractor associative memory ANNs*, designed to store and retrieve information in the form of neuronal firing patterns and/or sequences of neuronal firing patterns. Each pattern to be stored is represented as a microscopic state vector. One then constructs synapses and thresholds such that the dominant attractors of the network are precisely the pattern vectors (in

the case of static recall), or where, alternatively, they are trajectories in which the patterns are successively generated microscopic system states. From an initial configuration (the cue, or input pattern to be recognized) the system is allowed to evolve in time autonomously, and the final state (or trajectory) reached can be interpreted as the pattern (or pattern sequence) recognized by network from the input. For such programmes to work one clearly needs recurrent ANNs with extensive ergodicity breaking: the state vector will during the course of the dynamics (at least on finite time–scales) have to be confined to a restricted region of state–space (an ergodic component), the location of which is to depend strongly on the initial conditions. Hence our interest will mainly be in systems with many attractors. This, in turn, has implications at a theoretical/mathematical level: solving models of recurrent ANNs with extensively many attractors requires advanced tools from disordered systems theory, such as replica theory (statics) and generating functional analysis (dynamics).

Complex–Valued ANNs

It is expected that *complex–valued ANNs*, whose parameters (weights and threshold values) are all complex numbers, will have applications in all the fields dealing with complex numbers (e.g., telecommunications, quantum physics). A complex–valued, feedforward, multi–layered, back–propagation neural network model was proposed independently by T. Nitta [NF91, Nit97, Nit00, Nit04], G. GK92 [GK92] and N. Benvenuto [BP92, BP92], and demonstrated its characteristics:

(a) the properties greatly different from those of the real–valued back–propagation network, including 2D motion structure of weights and the orthogonality of the decision boundary of a complex–valued neuron;
(b) the learning property superior to the real–valued back–propagation;
(c) the inherent 2D motion learning ability (an ability to transform geometric figures); and
(d) the ability to solve the XOR problem and detection of symmetry problem with a single complex–valued neuron.

Following [NF91, Nit97, Nit00, Nit04], we consider here the complex–valued neuron. Its input signals, weights, thresholds and output signals are all complex numbers. The net input U_n to a complex–valued neuron n is defined as

$$U_n = W_{mn} X_m + V_n,$$

where W_{mn} is the (complex–valued) weight connecting the complex–valued neurons m and n, V_n is the (complex–valued) threshold value of the complex–valued neuron n, and X_m is the (complex–valued) input signal from the complex–valued neuron m. To get the (complex–valued) output signal, convert

the net input U_n into its real and imaginary parts as follows: $U_n = x + iy = z$, where $i = \sqrt{-1}$. The (complex–valued) output signal is defined to be

$$\sigma(z) = \tanh(x) + i \tanh(y),$$

where $\tanh(u) = (\exp(u) - \exp(-u)) = (\exp(u) + \exp(-u)),\ u \in \mathbb{R}$. Note that $-1 < \mathrm{Re}[\sigma],\ \mathrm{Im}[\sigma] < 1$. Note also that σ is not regular as a complex function, because the Cauchy–Riemann equations do not hold.

A complex–valued ANN consists of such complex–valued neurons described above. A typical network has 3 layers: $m \to n \to 1$, with $w_{ij} \in \mathbb{C}$ – the weight between the input neuron i and the hidden neuron j, $w_{0j} \in \mathbb{C}$ – the threshold of the hidden neuron j, $c_j \in \mathbb{C}$ – the weight between the hidden neuron j and the output neuron $(1 \le i \le m; 1 \le j \le n)$, and $c_0 \in \mathbb{C}$ – the threshold of the output neuron. Let $y_j(z), h(z)$ denote the output values of the hidden neuron j, and the output neuron for the input pattern $z = [z_1, \ldots, z_m]^t \in \mathbb{C}^m$, respectively. Let also $\nu_j(z)$ and $\mu(z)$ denote the net inputs to the hidden neuron j and the output neuron for the input pattern $z \in \mathbb{C}^m$, respectively. That is,

$$\begin{aligned} \nu_j(z) &= w_{ij} z_i + w_{0j}, & \mu(z) &= c_j y_j(z) + c_0, \\ y_j(z) &= \sigma(\nu_j(z)), & h(z) &= \sigma(\mu(z)). \end{aligned}$$

The set of all $m \to n \to 1$ complex–valued ANNs described above is usually denoted by $N_{m,n}$. The Complex–BP learning rule [NF91, Nit97, Nit00, Nit04] has been obtained by using a steepest–descent method for such (multilayered) complex–valued ANNs.

Common Continuous ANNs

Virtually all computer–implemented ANNs (mainly listed above) are discrete dynamical systems, mainly using supervised training (except Kohonen SOM) in one of gradient–descent searching forms. They are good as problem–solving tools, but they fail as models of animal nervous system. The other category of ANNs are continuous neural systems that can be considered as models of animal nervous system. However, *as models of the human brain, all current ANNs are simply trivial.*

Neurons as Functions

According to B. Kosko, neurons behave as functions [Kos92]; they transduce an unbounded input *activation* $x(t)$ into output *signal* $S(x(t))$. Usually a sigmoidal (S–shaped, bounded, monotone-nondecreasing: $S' \ge 0$) function describes the transduction, as well as the input–output behavior of many operational amplifiers. For example, the *logistic signal* (or, the *maximum–entropy*) function

$$S(x) = \frac{1}{1 + \mathrm{e}^{-cx}}$$

is sigmoidal and strictly increases for positive scaling constant $c > 0$. Strict monotonicity implies that the *activation derivative* of S is positive:

$$S' = \frac{dS}{dx} = cS(1 - S) > 0.$$

An infinitely steep logistic signal function gives rise to a threshold signal function

$$S(x^{n+1}) = \begin{cases} 1, & \text{if} \quad x^{n+1} > T, \\ S(x^n), & \text{if} \quad x^{n+1} = T, \\ 0, & \text{if} \quad x^{n+1} < T, \end{cases}$$

for an arbitrary real–valued threshold T. The index n indicates the discrete time step.

In practice signal values are usually binary or bipolar. *Binary signals*, like logistic, take values in the unit interval $[0, 1]$. *Bipolar signals* are signed; they take values in the bipolar interval $[-1, 1]$. Binary and bipolar signals transform into each other by simple scaling and translation. For example, the bipolar logistic signal function takes the form

$$S(x) = \frac{2}{1 + \mathrm{e}^{-cx}} - 1.$$

Neurons with bipolar threshold signal functions are called *McCulloch–Pits neurons*.

A naturally occurring bipolar signal function is the *hyperbolic–tangent* signal function

$$S(x) = \tanh(cx) = \frac{\mathrm{e}^{cx} - \mathrm{e}^{-cx}}{\mathrm{e}^{cx} + \mathrm{e}^{-cx}},$$

with activation derivative

$$S' = c(1 - S^2) > 0.$$

The *threshold linear* function is a binary signal function often used to approximate neuronal firing behavior:

$$S(x) = \begin{cases} 1, & \text{if} \quad cx \geq 1, \\ 0, & \text{if} \quad cx < 0, \\ cx, & \text{else}, \end{cases}$$

which we can rewrite as

$$S(x) = \min(1, \max(0, cx)).$$

Between its upper and lower bounds the threshold linear signal function is trivially monotone increasing, since $S' = c > 0$.

Gaussian, or bell–shaped, signal function of the form $S(x) = \mathrm{e}^{-cx^2}$, for $c > 0$, represents an important exception to signal monotonicity. Its activation derivative $S' = -2cx\mathrm{e}^{-cx^2}$ has the sign opposite the sign of the activation x.

Generalized Gaussian signal functions define potential or radial basis functions $S_i(x^i)$ given by

$$S_i(x) \;=\; \exp[-\frac{1}{2\sigma_i^2}\sum_{j=1}^{n}(x_j-\mu_j^i)^2],$$

for input activation vector $x=(x^i)\in\mathbb{R}^n$, variance σ_i^2, and mean vector $\boldsymbol{\mu}_i=(\mu_j^i)$. Each radial basis function S_i defines a spherical *receptive field* in $\mathbb{R}^n$. The ith neuron emits unity, or near-unity, signals for sample activation vectors x that fall in its receptive field. The mean vector $\boldsymbol{\mu}$ centers the receptive field in $\mathbb{R}^n$. The variance σ_i^2 localizes it. The radius of the Gaussian spherical receptive field shrinks as the variance σ_i^2 decreases. The receptive field approaches $\mathbb{R}^n$ as σ_i^2 approaches ∞.

The *signal velocity* $\dot{S}\equiv dS/dt$ is the *signal time derivative*, related to the activation derivative by

$$\dot{S} \;=\; S'\dot{x},$$

so it depends explicitly on *activation velocity*. This is used in unsupervised learning laws that adapt with *locally available information*.

The signal $S(x)$ induced by the activation x represents the neuron's firing frequency of action potentials, or pulses, in a sampling interval. The firing frequency equals the average number of pulses emitted in a sampling interval.

Short–term memory is modelled by *activation dynamics*, and *long–term memory* is modelled by *learning dynamics*. The overall neural network behaves as an *adaptive filter* (see [Hay91]).

In the simplest and most common case, neurons are not topologically ordered. They are related only by the synaptic connections between them. Kohonen calls this *lack of topological structure* in a *field of neurons* the *zeroth–order topology*. This suggests that ANN–models are *abstractions*, not *descriptions* of the brain neural networks, in which order does matter.

Basic Activation and Learning Dynamics

One of the oldest continuous training methods, based on Hebb's biological synaptic learning [Heb49], is *Oja–Hebb learning rule* [Oja82], which calculates the weight update according to the ODE

$$\dot{\omega}_i(t) = O(t)\,[I_i(t) - O(t)\,\omega_i(t)],$$

where $O(t)$ is the output of a simple, linear processing element; $I_i(t)$ are the inputs; and $\omega_i(t)$ are the synaptic weights.

Related to the Oja–Hebb rule is a special matrix of synaptic weights called *Karhunen–Loeve covariance matrix* $\mathbf{W}$ (KL), with entries

$$W_{ij} = \frac{1}{N}\omega_i^\mu\,\omega_j^\mu, \qquad \text{(summing over } \mu)$$

where N is the number of vectors, and ω_i^μ is the ith component of the μth vector. The KL matrix extracts the principal components, or directions of maximum information (correlation) from a dataset.

In general, continuous ANNs are *temporal dynamical systems*. They have two coupled dynamics: activation and learning. First, a general system of coupled ODEs for the output of the ith *processing element* (PE) x^i, called the *activation dynamics*, can be written as

$$\dot{x}^i = g_i(x^i, \underset{i}{\text{net}}), \tag{1.19}$$

with the *net input* to the ith PE x^i given by $\text{net}_i = \omega_{ij} x^j$.

For example,

$$\dot{x}^i = -x^i + f_i(\underset{i}{\text{net}}),$$

where f_i is called *output*, or *activation*, *function*. We apply some input values to the PE so that $\text{net}_i > 0$. If the inputs remain for a sufficiently long time, the output value will reach an equilibrium value, when $\dot{x}^i = 0$, given by $x^i = f_i(\text{net}_i)$. Once the unit has a nonzero output value, removal of the inputs will cause the output to return to zero. If $\text{net}_i = 0$, then $\dot{x}^i = -x^i$, which means that $x \to 0$.

Second, a general system of coupled ODEs for the *update* of the synaptic weights ω_{ij}, i.e, *learning dynamics*, can be written as a generalization of the Oja–Hebb rule, i.e..

$$\dot{\omega}_{ij} = G_i(\omega_{ij}, x^i, x^i),$$

where G_i represents the *learning law*; the learning process consists of finding weights that encode the knowledge that we want the system to learn. For most realistic systems, it is not easy to determine a closed–form solution for this system of equations, so the approximative solutions are usually enough.

Standard Models of Continuous Nets

Hopfield Continuous Net

One of the first physically–based ANNs was developed by J. Hopfield. He first made a discrete, Ising–spin based network in [Hop82], and later generalized it to the continuous, graded–response network in [Hop84], which we briefly describe here. Later we will give full description of Hopfield models. Let $\text{net}_i = u_i$ – the net input to the ith PE, biologically representing the summed action potentials at the axon hillock of a neuron. The PE *output function* is

$$v_i = g_i(\lambda u_i) = \frac{1}{2}(1 + \tanh(\lambda u_i)),$$

where λ is a constant called the *gain parameter*. The network is described as a transient RC circuit

$$C_i \dot{u}_i = T_{ij} v_j - \frac{u_i}{R_i} + I_i, \tag{1.20}$$

where I_i, R_i and C_i are inputs (currents), resistances and capacitances, and T_{ij} are synaptic weights.

The Hamiltonian energy function corresponding to (1.20) is given as

$$H = -\frac{1}{2}T_{ij}v_iv_j + \frac{1}{\lambda}\frac{1}{R_i}\int_0^{v_i} g_i^{-1}(v)\,dv - I_iv_i, \qquad (j \neq i) \tag{1.21}$$

which is a generalization of a discrete, *Ising–spin Hopfield network* with energy function

$$E = -\frac{1}{2}\omega_{ij}x^ix^j, \qquad (j \neq i).$$

where $g_i^{-1}(v) = u$ is the inverse of the function $v = g(u)$. To show that (1.21) is an appropriate *Lyapunov function* for the system, we shall take its time derivative assuming T_{ij} are symmetric:

$$\dot{H} \;=\; -\dot{v}_i(T_{ij}v_j - \frac{u_i}{R_i} + I_i) \;=\; -C_i\dot{v}_i\dot{u}_i \;=\; -C_i\dot{v}_i^2\frac{\partial g_i^{-1}(v_i)}{\partial v_i}. \tag{1.22}$$

All the factors in the summation (1.22) are positive, so $\dot{H}$ must decrease as the system evolves, until it eventually reaches the stable configuration, where $\dot{H} = \dot{v}_i = 0$.

Hecht–Nielsen Counterpropagation Net

Hecht–Nielsen counterpropagation network (CPN) is a full–connectivity, graded–response generalization of the standard BP algorithm (see [Hec87, Hec90]). The outputs of the PEs in CPN are governed by the set of ODEs

$$\dot{x}^i = -Ax_i + (B - x^i)I_i - x^i\sum_{j\neq i} I_j,$$

where $0 < x^i(0) < B$, and $A, B > 0$. Each PE receives a net excitation (on–center) of $(B - x^i)I_i$ from its corresponding input value, I. The addition of inhibitory connections (off–surround), $-x^iI_j$, from other units is responsible for preventing the activity of the processing element from rising in proportion to the absolute pattern intensity, I_i. Once an input pattern is applied, the PEs quickly reach an equilibrium state ($\dot{x}^i = 0$) with

$$x^i = \Theta_i\frac{BI_i}{A + I_i},$$

with the normalized *reflectance pattern* $\Theta_i = I_i\left(\sum_i I_i\right)^{-1}$, such that $\sum_i \Theta_i = 1$.

Competitive Net

Activation dynamics is governed by the ODEs

$$\dot{x}^i = -Ax_i + (B - x^i)[f(x^i) + \underset{i}{\text{net}}] - x^i \left[\sum_{j \neq i} f(x_j) + \sum_{j \neq i} \text{net}_j \right],$$

where $A, B > 0$ and $f(x^i)$ is an output function.

Kohonen's Continuous SOM and Adaptive Robotics Control

Kohonen continuous self organizing map (SOM) is actually the original Kohonen model of the biological neural process (see [Koh88]). SOM activation dynamics is governed by

$$\dot{x}^i = -r_i(x^i) + \underset{i}{\text{net}} + z_{ij} x_j, \tag{1.23}$$

where the function $r_i(x^i)$ is a general form of a loss term, while the final term models the lateral interactions between units (the sum extends over all units in the system). If z_{ij} takes the form of the Mexican–hat function, then the network will exhibit a bubble of activity around the unit with the largest value of net input.

SOM learning dynamics is governed by

$$\dot{\omega}_{ij} = \alpha(t)(I_i - \omega_{ij})U(x^i),$$

where $\alpha(t)$ is the learning momentum, while the function $U(x^i) = 0$ unless $x^i > 0$ in which case $U(x^i) = 1$, ensuring that only those units with positive activity participate in the learning process.

Kohonen's continuous SOM (1.23–1.2.2) is widely used in adaptive robotics control. Having an $n-$segment robot arm with n chained $SO(2)-$joints, for a particular initial position x and desired velocity $\dot{x}^j_{desir}$ of the end–effector, the required torques T_i in the joints can be found as

$$T_i = a_{ij}\, \dot{x}^j_{desir},$$

where the inertia matrix $a_{ij} = a_{ij}(x)$ is learned using SOM.

Adaptive Resonance Theory

Principles derived from an analysis of experimental literatures in vision, speech, cortical development, and reinforcement learning, including attentional blocking and cognitive–emotional interactions, led to the introduction of S. Grossberg's *adaptive resonance theory* (ART) as a theory of human *cognitive information processing* (see [CG03]). The theory has evolved as a series

of real–time neural network models that perform unsupervised and supervised learning, pattern recognition, and prediction. Models of unsupervised learning include ART1, for binary input patterns, and fuzzy–ART and ART2, for analog input patterns [Gro82, CG03]. ARTMAP models combine two unsupervised modules to carry out supervised learning. Many variations of the basic supervised and unsupervised networks have since been adapted for technological applications and biological analyzes.

A central feature of all ART systems is a *pattern matching process* that compares an external input with the internal memory of an active code. ART matching leads either to a resonant state, which persists long enough to permit learning, or to a parallel memory search. If the search ends at an established code, the memory representation may either remain the same or incorporate new information from matched portions of the current input. If the search ends at a new code, the memory representation learns the current input. This match–based learning process is the foundation of ART *code stability*. Match–based learning allows memories to change only when input from the external world is close enough to internal expectations, or when something completely new occurs. This feature makes ART systems well suited to problems that require on–line learning of large and evolving databases (see [CG03]).

Many ART applications use fast learning, whereby adaptive weights converge to equilibrium in response to each input pattern. Fast learning enables a system to adapt quickly to inputs that occur rarely but that may require immediate accurate recall. Remembering details of an exciting movie is a typical example of learning on one trial. Fast learning creates memories that depend upon the order of input presentation. Many ART applications exploit this feature to improve accuracy by voting across several trained networks, with voters providing a measure of confidence in each prediction.

Match–based learning is complementary to *error–based learning*, which responds to a mismatch by changing memories so as to reduce the difference between a target output and an actual output, rather than by searching for a better match. Error–based learning is naturally suited to problems such as adaptive control and the learning of *sensory–motor maps*, which require ongoing adaptation to present statistics. Neural networks that employ error–based learning include backpropagation and other multilayer perceptrons (MLPs).

Activation dynamics of ART2 is governed by the ODEs [Gro82, CG03]

$$\epsilon \dot{x}_i = -Ax_i + (1 - Bx_i)I_i^+ - (C + Dx_i)I_i^-,$$

where ϵ is the 'small parameter', I_i^+ and I_i^- are excitatory and inhibitory inputs to the ith unit, respectively, and $A, B, C, D > 0$ are parameters.

General *Cohen–Grossberg activation equations* have the form:

$$\dot{v}_j = -a_j(v_j)[b_j(v_j) - f_k(v_k)m_{jk}], \qquad (j = 1, \ldots, N), \tag{1.24}$$

and the *Cohen–Grossberg theorem* ensures the global stability of the system (1.24). If

$$a_j = 1/C_j, b_j = v_j/R_j - I_j, f_j(v_j) = u_j,$$

and constant $m_{ij} = m_{ji} = T_{ji}$, the system (1.24) reduces to the Hopfield circuit model (1.20).

ART and distributed ART (dART) systems are part of a growing family of self–organizing network models that feature attentional feedback and stable code learning. Areas of technological application include industrial design and manufacturing, the control of mobile robots, face recognition, remote sensing land cover classification, target recognition, medical diagnosis, electrocardiogram analysis, signature verification, tool failure monitoring, chemical analysis, circuit design, protein/DNA analysis, 3D visual object recognition, musical analysis, and seismic, sonar, and radar recognition. ART principles have further helped explain parametric behavioral and brain data in the areas of visual perception, object recognition, auditory source identification, variable–rate speech and word recognition, and *adaptive sensory–motor control* (see [CG03]).

Spatiotemporal Networks

In *spatiotemporal networks*, activation dynamics is governed by the ODEs

$$\begin{aligned}
\dot{x}^i &= A(-ax_i + b[I_i - \Gamma]^+), \\
\dot{\Gamma} &= \alpha(S - T) + \beta\dot{S}, \qquad \text{with} \\
[u]^+ &= \begin{cases} u & \text{if} \quad u > 0 \\ 0 & \text{if} \quad u \le 0 \end{cases}, \\
A(u) &= \begin{cases} u & \text{if} \quad u > 0 \\ cu & \text{if} \quad u \le 0 \end{cases}.
\end{aligned}$$

where $a, b, \alpha, \beta > 0$ are parameters, $T > 0$ is the *power–level target*, $S = \sum_i x^i$, and $A(u)$ is called the *attack function*.

Learning dynamics is given by *differential Hebbian law*

$$\begin{aligned}
\dot{\omega}_{ij} &= (-c\omega_{ij} + dx_i x_j)U(\dot{x}^i)U(-\dot{x}_j), \qquad \text{with} \\
U(s) &= \begin{cases} 1 & \text{if} \quad s > 0 \\ 0 & \text{if} \quad s \le 0 \end{cases} \qquad \text{where } c, d > 0 \text{ are constants.}
\end{aligned}$$

Fuzzy Systems

Recall that f*uzzy expert systems* are based on *fuzzy logic* (FL), which is itself derived from *fuzzy set theory* dealing with reasoning that is approximate

rather than precisely deduced from classical *predicate logic*.[150] FL, introduced in 1965 by Prof. Lotfi Zadeh at the University of California, Berkeley, can be thought of as the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem.[151] FL allows for set membership values between and including 0 and 1, shades of gray as well as black and white, and in its linguistic form, imprecise concepts like 'slightly', 'quite' and 'very'. Specifically, it allows partial membership in a set. It is related to fuzzy sets and possibility theory.

[150] Recall that *predicate* or *propositional logic* (PL) is a system for evaluating the validity of arguments by encoding them into sentential variables and boolean operator and is part of the philosophy of *formal logic*. The actual truth of the *premises* is not particularly relevant in PL; it is dealing mostly with the structure of an argument so that if it so happens that the premises are true, the conclusion either must be true, or could perhaps be false. If it is demonstrable that the conclusion must be true then the original argument can be said to be valid. However, if it is possible for all of the premises to be true, and yet still have a false conclusion, the sequent is invalid. In an ordinary PL, there is one unitary operator, four binary operators and two quantifiers. The only unary operator in PL is the negation, usually denoted by $\neg P$, which is the opposite of the predicate (i.e., Boolean variable) P. The binary operators are: (i) *conjunction* $\wedge$, which is true iff both of the Boolean conjuncts are true; (ii) *disjunction* $\vee$, which is false iff both of the Boolean disjuncts are false; (iii) implication (or, conditional), meaning, *if* P *then* Q, and denoted $P \Longrightarrow Q$, where P is *antecedent* and Q is *consequent*; implication is false only iff from true P follows false Q; (iv) equivalence, or bi–conditional is a double–sided implication, $(P \Longrightarrow Q) \wedge (Q \Longrightarrow P)$; it is false iff from true P follows false Q and from true Q follows false P. Besides, PL also has the *universal quantifier* $\forall$, meaning 'for al', and the *existential quantifier* $\exists$, meaning 'there is'.

[151] Note that *degrees of truth* in fuzzy logic are often confused with probabilities. However, they are conceptually distinct; fuzzy truth represents membership in vaguely defined sets, not likelihood of some event or condition. To illustrate the difference, consider this scenario: Bob is in a house with two adjacent rooms: the kitchen and the dining room. In many cases, Bob's status within the set of things 'in the kitchen' is completely plain: he's either 'in the kitchen' or 'not in the kitchen'. What about when Bob stands in the doorway? He may be considered 'partially in the kitchen'. Quantifying this partial state yields a fuzzy set membership. With only his big toe in the dining room, we might say Bob is 99% 'in the kitchen' and 1% 'in the dining room', for instance. No event (like a coin toss) will resolve Bob to being completely 'in the kitchen' or 'not in the kitchen', as long as he's standing in that doorway. Fuzzy sets are based on vague definitions of sets, not randomness. Fuzzy logic is controversial in some circles, despite wide acceptance and a broad track record of successful applications. It is rejected by some control engineers for validation and other reasons, and by some statisticians who hold that probability is the only rigorous mathematical description of uncertainty. Critics also argue that it cannot be a superset of ordinary set theory since membership functions are defined in terms of conventional sets.

'Fuzzy Thinking'

'There is no logic in logic', pronounced the father of fuzzy logic, Lotfi Zadeh. His cryptic play–on–words, he explained, means that the kind of logic that people use to solve most real world problems rather than the artificial problems for which mathematical solutions are available is not the kind of logic that engineers are taught in school. 'An engineer can solve problems throughout his whole career without ever needing to resort to the brand of logic he was trained in', said Zadeh. 'Why? Because all people, even engineers, compute with words not the logical symbols taught in school', Zadeh maintained. 'In the future, computing will be done with words from natural languages, rather than with symbols that are far removed from daily life.'

In 1973, Zadeh proposed the concept of linguistic or fuzzy variables [Zad65, Zad78, Yag87]. Think of them as linguistic objects or words, rather than numbers. The sensor input is a noun, e.g., temperature, displacement, velocity, ow, pressure, etc. Since error is just the difference, it can be thought of the same way. The fuzzy variables themselves are adjectives that modify the variable (e.g., large positive error, small positive error, zero error, small negative error, and large negative error). As a minimum, one could simply have positive, zero, and negative variables for each of the parameters.

Additional ranges such as very large and very small could also be added to extend the responsiveness to exceptional or very nonlinear conditions, but are not necessary in a basic system. Normal logic is just not up to modelling the real world, claims Bart Kosko [Kos92, Kos93, Kos96, Kos99], perhaps the worlds most active proponent of fuzzy logic. According to Kosko, there is always *ambiguity* in our perceptions and measurements that is difficult to reflect in traditional logic. Probability attempts to reflect ambiguity by resorting to statistical averages over many events. But fuzzy theory describes the ambiguity of individual events. It measures the degree to which an event occurs, not whether it occurs.

Fuzzy Sets

Recall that a crisp (ordinary mathematical) set X is defined by a binary characteristic function $\chi_X(x)$ of its elements x

$$\chi_X(x) = \begin{cases} 1, & \text{if } \ x \in X, \\ 0, & \text{if } \ x \notin X, \end{cases}$$

while a fuzzy set is defined by a continuous characteristic function

$$\chi_X(x) = [0, 1]\,,$$

including all (possible) real values between the two crisp extremes 1 and 0, and including them as special cases.

More precisely, a fuzzy set X is defined as a collection of ordered pairs

$$X = \{(x, \mu(x))\}, \tag{1.25}$$

where $\mu(x)$ is the *fuzzy membership function* representing the grade of membership of the element x in the set X. A single pair is called a fuzzy *singleton*.

Lotfi Zadeh claimed that many *sets* in the world that surrounds us are defined by a non-distinct boundary. Indeed, the *set of high mountains* is an example of such sets. Zadeh decided to extend two–valued logic, defined by the binary pair $\{0, 1\}$ to the whole continuous interval $[0, 1]$ thereby introducing a gradual transition from falsehood to truth. The original and pioneering papers on fuzzy sets by Zadeh [Zad65, Zad78, Yag87] explain the theory of fuzzy sets that result from the extension as well as a fuzzy logic based on the set theory.

Fuzzy sets are a further development of the mathematical concept of a set. Sets were first studied formally by German mathematician Georg Cantor (1845–1918). His theory of sets met much resistance during his lifetime, but nowadays most mathematicians believe it is possible to express most, if not all, of mathematics in the language of set theory. Many researchers are looking at the consequences of 'fuzzifying' set theory, and much mathematical literature is the result.

Conventional sets. A set is any collection of objects which can be treated as a whole. Cantor described a set by its members, such that an item from a given universe is either a member or not. Almost anything called a *set* in ordinary conversation is an acceptable set in the mathematical sense. A set can be specified by its members, they characterize a set completely. The list of members $A = \{0, 1, 2, 3\}$ specifies a finite set. Nobody can list all elements of an *infinite set*, we must instead state some property which characterizes the elements in the set, for instance the predicate $x > 10$. That set is defined by the elements of the *universe of discourse* which make the predicate true. So there are two ways to describe a set: explicitly in a list or implicitly with a predicate.

Fuzzy sets. Following Zadeh many sets have more than an *Either–Or* criterion for membership. Take for example the set of *young people*. A one year old baby will clearly be a member of the set, and a 100 years old person will not be a member of this set, but what about people at the age of 20, 30, or 40 years? Another example is a weather report regarding high temperatures, strong winds, or nice days. In other cases a criterion appears nonfuzzy, but is perceived as fuzzy: a speed limit of 60 kilometers per hour, a check–out time at 12 noon in a hotel, a 50 years old man. Zadeh proposed a *grade of membership*, such that the transition from membership to non–membership is gradual rather than abrupt.

The grade of membership for all its members thus describes a fuzzy set. An item's grade of membership is normally a real number between 0 and 1, often denoted by the Greek letter μ. The higher the number, the higher the

membership. Zadeh regards Cantor's set as a special case where elements have full membership, i.e., $\mu = 1$. He nevertheless called Cantor's sets *nonfuzzy*; today the term *crisp* set is used, which avoids that little dilemma.

The membership for a 50 year old in the set *young* depends on one's own view. The grade of membership is a precise, but subjective measure that depends on the context.

A fuzzy membership function is different from a statistical probability distribution. A possible event does not imply that it is probable. However, if it is probable it must also be possible. We might view a fuzzy membership function as our personal distribution, in contrast with a statistical distribution based on observations.

Universe of discourse. Elements of a fuzzy set are taken from a *universe of discourse*. It contains all elements that can come into consideration. Even the universe of discourse depends on the context. An application of the universe is to suppress faulty measurement data. In case we are dealing with a non–numerical quantity, for instance *taste*, which cannot be measured against a numerical scale, we cannot use a numerical universe. The elements are then said to be taken from a *psychological continuum*.

Membership Functions. Every element in the universe of discourse is a member of the fuzzy set to some grade, maybe even zero. The set of elements that have a non–zero membership is called the *support* of the fuzzy set. The function that ties a number to each element $x$of the universe is called the *membership function*.

Continuous and discrete representations. There are two alternative ways to represent a membership function in a computer: continuous or discrete. In the continuous form the membership function is a mathematical function, possibly a program. A membership function is for example bell-shaped (also called a $\pi-curve$), $s-$shaped (called an $s-curve$), a reverse $s-curve$ (called $z-curve$), triangular, or trapezoidal. In the discrete form the membership function and the universe are discrete points in a list (vector). Sometimes it can be more convenient with a sampled (discrete) representation. As a very crude rule of thumb, the continuous form is more CPU intensive, but less storage demanding than the discrete form.

Normalization. A fuzzy set is *normalized* if its largest membership value equals 1. We normalize by dividing each membership value by the largest membership in the set, $a/max(a)$.

Singletons. Strictly speaking, a fuzzy set A is a collection of ordered pairs: $A = \{(x, \mu(x))\}$.

Item x belongs to the universe and $\mu(x)$ is its *grade of membership* in A. A single pair $(x, \mu(x))$ is called a fuzzy *singleton*; thus the whole set can be viewed as the union of its constituent singletons.

Linguistic variables. Just like an algebraic variable takes numbers as values, a *linguistic variable* takes words or sentences as values [Yag87, Kos92]. The set of values that it can take is called its *term set*. Each value in the term set is a *fuzzy variable* defined over a *base variable*. The base variable defines the universe of discourse for all the fuzzy variables in the term set. In short, the hierarchy is as follows:

$$\text{linguistic variable} \;\rightarrow\; \text{fuzzy variable} \;\rightarrow\; \text{base variable.}$$

Primary terms. A *primary term* is a term or a set that must be defined a priori, for example *Young* and *Old*, whereas the sets *Very Young* and *Not Young* are modified sets.

Fuzzy set operations. A fuzzy *set operation* creates a new set from one or several given sets.

Let A and B be fuzzy sets on a mutual universe of discourse X. If these were ordinary (crisp) sets, we would have the following definitions:

The *intersection* of A and B is: $A \cap B \equiv \min\{A, B\}$, where *min* is an item–by–item minimum operation.

The *union* of A and B is: $A \cup B \equiv \max\{A, B\}$, where *max* is an item–by–item maximum operation.

The *complement* of A is: $\neg A \equiv 1 - A$, where in a each membership value is subtracted from 1.

However, as A and B are fuzzy sets, the following definitions are more appropriate:

The *intersection* of A and B is: $A \cap B \equiv \min\{\mu_A(X), \mu_B(X)\}$, where *min* is an item–by–item minimum operation.

The *union* of A and B is: $A \cup B \equiv \max\{\mu_A(X), \mu_B(X)\}$, where *max* is an item–by–item maximum operation.

The *complement* of A is: $\neg A \equiv 1 - \mu_A(X)$, where in a each membership value is subtracted from 1.

Fuzzy Example

Using fuzzy membership functions $\mu(x)$, we can express both physical and non–physical quantities (e.g., temperature, see Figure 1.29) using *linguistic variables*.

Various logical combinations of such linguistic variables leads to the concept of fuzzy–logic control. Recall that basic logical operations AND, OR, NOT are defined as:

$$\begin{aligned} AND: &\quad C \cap W \quad - \quad \text{intersection of crisp sets} \quad C, W, \\ OR: &\quad C \cup W \quad - \quad \text{union of crisp sets} \quad C, W, \\ NOT: &\quad \neg C \quad - \quad \text{complement of a crisp set} \quad C. \end{aligned}$$

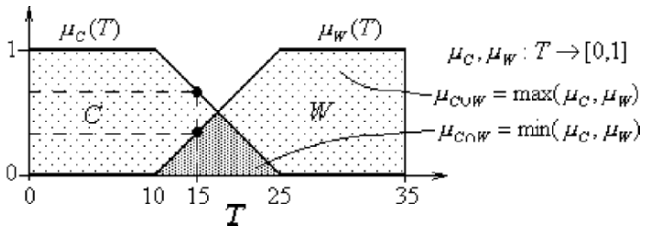

Fig. 1.29. Fuzzy–set description of *cold* (C) and *warm* (W) *temperature* (T), using the membership functions $\mu_C(T)$ and $\mu_W(T)$, respectively. For example, fuzzy answers to the questions "How cold is 15°?" and "How warm is 15°?" are given by: "15° *is quite cold* as $\mu_C(15) = 2/3$" and "15° *is not really warm* as $\mu_W(15) = 1/3$", respectively.

The corresponding fuzzy–logic operations are defined as:

$$\begin{aligned} AND: &\quad \mu_{C\cap W}(T) = \min\{\mu_C(T), \mu_W(T)\}, \\ OR: &\quad \mu_{C\cup W}(T) = \max\{\mu_C(T), \mu_W(T)\}, \\ NOT: &\quad \mu_{\neg C}(T) = 1 - \mu_C(T). \end{aligned}$$

Fuzziness of the Real World

The real world consists of all subsets of the universe and the only subsets that are not fuzzy are the constructs of classical mathematics.

From small errors to satisfied customers to safe investments to noisy signals to charged particles, each element of the real world is in some measure fuzzy. For instance, satisfied customers can be somewhat unsatisfied, safe investments somewhat unsafe and so on. What is worse, most events more or less smoothly transition into their opposites, making classification difficult near the midpoint of the transition. Unfortunately, textbook events and their opposites are crisp, unlike the real world. Take the proposition that there is a 50% chance that an apple is in the refrigerator. That is an assertion of crisp logic. But suppose upon investigation it is found that there is half an apple in the refrigerator, that is fuzzy.

But regardless of the realities, the crisp logic in vogue today assumes that the world is really unambiguous and that the only uncertainty is the result of random samples from large sets. As the facts about these large sets become better known, the randomness supposedly dissipates, so that if science had access to all the facts, it would disappear. Unfortunately, if all the facts were in, a platypus would remain only roughly an mammal.

On the other hand, fuzzy logic holds that uncertainty is deterministic and does not dissipate as more elements of a set are examined. Take an ellipse, for instance. It is approximately a circle, to whatever degree that it resembles a perfect circle. There is nothing random about it. No matter how precisely it is measured it remains only approximately a circle. All the facts are in and yet uncertainty remains.

Traditional crisp logic has a difficult time applying itself to very large sets, since probability fades to unity, as well as to individual events where probabilities cannot be defined at all. Nevertheless, crisp logic continues to rein supreme based on long standing western traditions that maintain that rationality would vanish if there were not crisp logical ideals to which we should aspire. These laws of (rational) thought were first characterized by Aristotle as the principle of non-contradiction and the principle of the excluded middle. The principle of non-contradiction, stated in words, says that nothing can be both A and $\neg A$. The law of the excluded middle says that anything must be either A or $\neg A$.

'Fuzziness is the denial of both these so–called laws', says E. Cox [Cox92, Cox94]). The classical example is of a platypus which both is and is not a mammal. In such individual cases, even appending probability theory to crisp logic cannot resolve the paradox. For instance, take the now classical paradox formulated by B. Russell: If a barber shaves everyone in a village who does not shave himself, then who shaves the barber? This paradox was devised to assault G. Cantor's set theory as the foundation for G. Boole's digital logic. It has been restated in many forms, such as the liar from Crete who said that all Creatans are liars. Russell solved it by merely disqualifying such self–referential statements in his set theory. Probability theory solves it by assuming a population of barbers 50% of whom do, and 50% of whom do not, shave themselves. But fuzzy logic solves it by assigning to this individual barber a 50% membership value in the set self-shaving barbers. Further, it shows that there is a whole spectrum of other situations that are less fuzzy and which correspond to other degrees of set membership. Such as, barbers who shave themselves 70% of the time.

Kosko illustrates these various degrees of ambiguity by geometrically plotting various degrees of set membership inside a *unit fuzzy hypercube* $[0,1]^n$ [Kos92, Kos93, Kos96, Kos99]. This sets–as–points approach holds that a fuzzy set is a point in a unit hypercube and a non–fuzzy set is a corner of the hypercube. Normal engineering practice often visualizes binary logical values as the corners of a hypercube, but only fuzzy theory uses the inside of the cube. Fuzzy logic is a natural filling–in of traditional set theory. Any engineer will recognize the 3D representation of all possible combinations three Boolean values: $\{0,0,0\}, \{0,0,1\}, \{0,1,0\}, \{0,1,1\}, \{1,0,0\}, \{1,0,1\}, \{1,1,0\}, \{1,1,1\}$, which correspond to the corners of the unit hypercube. But fuzzy logic also allows any other fractional values inside the hypercube, such as $\{0.5, 0.7, 0.3\}$ corresponding to degrees of set membership.

Fuzzy logic holds that any point inside the unit hypercube is a fuzzy set with Russell's paradox located at the point of maximum ambiguity in the center of the hypercube.

Fuzzy Entropy

Degrees of fuzziness are referred to as entropy by Kosko. *Fuzzy mutual entropy* measures the *ambiguity of a situation*, information and entropy are inversely

related – if you have a maximum–entropy solution, then you have a minimum–information solution, and visa versa, according to Kosko. But minimum–information does not mean that too little information is being used. On the contrary, the principle of maximum entropy ensures that only the relevant information is being used.

This idea of maximizing entropy, according to Kosko, is present throughout the sciences, although it is called by different names. 'From the quantum level up to astrophysics or anywhere in–between for pattern recognition, you want to use all and only the available information,' Kosko claims. This emergent model proposes that scientists and engineers estimate the uncertainty structure of a given environment and maximize the entropy relative to the known information, similar to the Lagrange technique in mathematics. The principle of maximum entropy states that any other technique has to be biased, because it has less entropy and thus uses more information than is really available.

Fuzzy theory provides a measure of this entropy factor. It measures ambiguity with operations of union $\cup$, intersection $\cap$ and complement $\neg$.

In traditional logic, these three operators are used to define a set of axioms that were proposed by Aristotle to be the immutable laws of (rational) thought, namely, the principle of *non–contradiction* and the principle of the *excluded middle*. The principle of non–contradiction, that nothing can be both A and $\neg A$, and the law of the excluded middle, that anything must be either A or $\neg A$, amounts to saying that the intersection of a set and its complement is always empty and that the union of a set and its complement always equals the whole *universe of discourse*, respectively. But if we do not know A with certainty, then we do not know $-A$ with certainty either, else by double negation we would know A with certainty. This produces non–degenerate *overlap* $(A \cap \neg A)$, which breaks the law of non–contradiction. Equivalently, it also produced non–degenerate *underlap* $(A \cup \neg A)$ which breaks the law of the excluded middle. In fuzzy logic both these so–called laws are denied. A set and its complement can both be overlap and underlap.

What is worse, there is usually ambiguity in more than one parameter or dimension of a problem. To represent multi–dimensional ambiguity, Kosko shows fuzzy entropy geometrically with a hypercube.

All these relationships are needed in fuzzy logic to express its basic structures for *addition*, *multiplication*, and most important, *implication* $IF \Rightarrow THEN$. They all follow from the subsethood relationships between fuzzy sets. The subset relation by itself, corresponds to the implication relation in crisp logic. For instance, $A \Rightarrow B$ is *false only* if the *antecedent* A is *true* and the *consequent* B is *false*. The same holds for subsets, A is a subset of B if there is no element that belongs to A but not to B.

But in fuzzy logic, degrees of subsethood permit some A to be somewhat of a subset of B even though some of its elements are not elements of B. The degree to which A is a subset of B can be measured as the distance from the origin to $(A \cap B)$ divided by the distance from the origin to A.

This structure is derived as a theorem of fuzzy logic, whereas for probability theory equivalent conditional probability theorem has to be assumed, making fuzzy logic a more fundamental.

The *fuzzy mutual entropy* measures how close a fuzzy description of the world is to its own opposite [Kos99]. It has no random analogue in general. The *fuzzy fluid* leads to a type of wave equation. The wave shows how the *extended Shannon entropy potential* $S : [0,1]^n \to \mathbb{R}$, defined on the *entire fuzzy cube* $[0,1]^n$, fluctuates in time. It has the form of a *reaction–diffusion* equation

$$\dot{S} = -c\,\nabla^2 S, \tag{1.26}$$

where c is the *fuzzy diffusion parameter*. The *fuzzy wave equation* (1.26) implies $\dot{S} > 0$, and thus resembles the entropy increase of the $S-$theorem of the *Second Law of thermodynamics*.

Similar equations occur in all branches of science and engineering. The Schrödinger wave equation (see [II06a, II06b]) has this form, as well as most models of diffusion. The fuzzy wave equation (1.26) assumes only that information is conserved. The total amount of information is fixed and we do not create or destroy information. Some form of the wave equation would still apply if information were conserved locally or in small regions of system space. The space itself is a fuzzy cube of high dimension. It has as many dimensions as there are objects of interest. The Shannon entropy S changes at each point in this cube and defines a *fuzzy wave*. The obvious result is that the entropy S can only grow in time in the spirit of the second law.

The entropy always grows but its *rate of growth* depends on the system's position in the *fuzzy parameter space*. A deeper result is that entropy changes slowest at the fuzzy cube *midpoint of maximum fuzz*. That is the only point in the cube where the fuzzy description equals its own opposite. The Shannon entropy wave grows faster and faster away from the cube midpoint and near its skin. The skin or surface of the fuzzy cube is the only place where a 0 or 1 appears in the system description. The fuzzy wave equation (1.26) shows that the entropy S changes infinitely fast iff it touches the cubes's skin. However, this is impossible in a universe with finite bounds on velocity like the speed of light. So, the result is never a *bit* – it is always a *fit* [Kos99].

Fuzzy Patches for System Modelling

Like ANNs, the fuzzy logic systems are generic *function approximators* [Kos92]. Namely, fuzzy system modelling is performed as a *nonlinear function approximation* using the so–called *fuzzy patches* (see Figure 1.30), which approximate the given function $y = f(x)$, i.e., the *system input–output relation*. The fuzzy patches R_i are given by a set of canonical fuzzy IF–THEN rules:

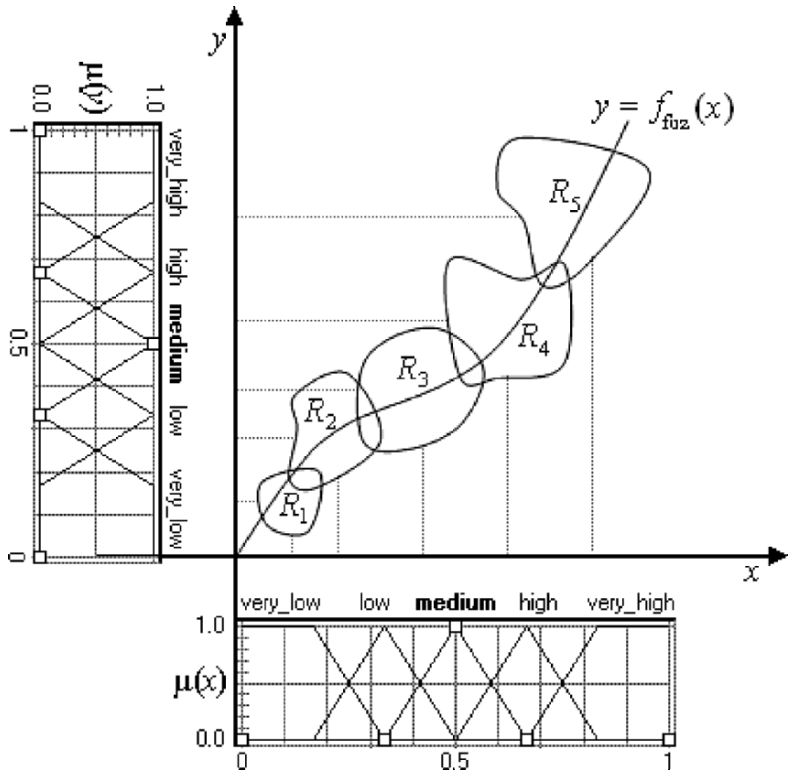

Fig. 1.30. Fuzzy–logic approximation $y = f_{\text{fuz}}(x)$ of an arbitrary function $y = f(x)$ using *fuzzy patches* R_i given by a set of canonical fuzzy IF–THEN rules.

$$
\begin{aligned}
R_1 &: IF \quad x \quad is \quad A_1 \quad THEN \quad y \quad is \quad R_1, \\
R_2 &: IF \quad x \quad is \quad A_2 \quad THEN \quad y \quad is \quad R_2, \\
&\vdots \\
R_n &: IF \quad x \quad is \quad A_n \quad THEN \quad y \quad is \quad R_n.
\end{aligned}
$$

Fuzzy Inference Engine

In the realm of fuzzy logic the above generic nonlinear function approximation is performed by means of fuzzy inference engine. The *fuzzy inference engine* is an *input–output dynamical system* which *maps* a set of input linguistic variables ($IF-$part) into a set of output linguistic variables ($THEN-$part). It has three sequential modules (see Figure 1.31):

1. *Fuzzification*; in this module numerical crisp input variables are fuzzified; this is performed as an overlapping partition of their universes of discourse by means of fuzzy membership functions $\mu(x)$ (1.25), which can have

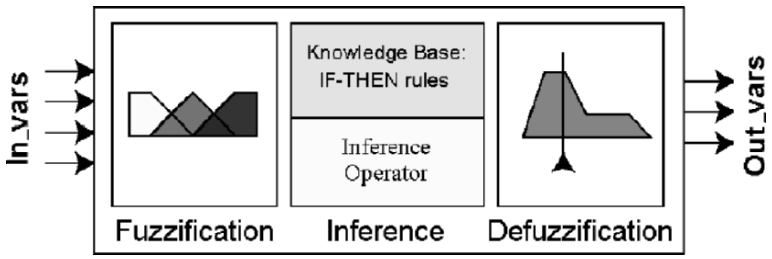

Fig. 1.31. Basic structure of the fuzzy inference engine.

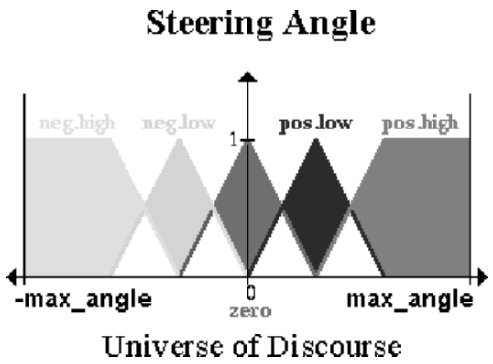

Fig. 1.32. Fuzzification example: set of triangular–trapezoidal membership functions partitioning the universe of discourse for the angle of the hypothetical steering wheel; notice the white overlapping triangles.

various shapes, like triangular–trapezoidal (see Figure 1.32), Gaussian–bell, $\mu(x) = \exp\left[\frac{-(x-m)^2}{2\sigma^2}\right]$ (with mean m and standard deviation σ), sigmoid $\mu(x) = \left[1 + \left(\frac{x-m}{\sigma}\right)^2\right]^{-1}$, or some other shapes.

B. Kosko and his students have done extensive computer simulations looking for the best shape of fuzzy sets to model a known test system as closely as possible. They let fuzzy sets of all shapes and sizes compete against each other. They also let neural systems tune the fuzzy–set curves to improve how well they model the test system. The main conclusion from these experiments is that 'triangles never do well' in such contests. Suppose we want an adaptive fuzzy system $F : \mathbb{R}^n \to \mathbb{R}$ to approximate a test function (or, approximand) $f : \mathbb{R}^n \to \mathbb{R}$ as closely as possible in the sense of minimizing the mean–squared error between them, $\left(\|f - F\|^2\right)$. Then the ith scalar 'sinc' function (as commonly used in signal processing),

$$\mu_i(x) = \frac{\sin\left(\frac{x-m_i}{d_i}\right)}{\frac{x-m_i}{d_i}}, \qquad (i = 1, \ldots, n), \tag{1.27}$$

with center m_i and dispersion (width) $d_i = \sigma_i^2 > 0$, often gives the best performance for $IF-$part mean–squared function approximation, even though this generalized function can take on negative values (see [Kos99]).

2. *Inference*; this module has two submodules:
   (i) The expert–knowledge base consisting of a set of $IF-THEN$ rules relating input and output variables, and
   (ii) The inference method, or implication operator, that actually combines the rules to give the fuzzy output; the most common is *Mamdani Min–Max inference*, in which the membership functions for input variables are first combined inside the $IF-THEN$ rules using AND ($\cap$, or Min) operator, and then the output fuzzy sets from different $IF-THEN$ rules are combined using OR ($\cup$, or Max) operator to get the common fuzzy output (see Figure 1.33).
3. *Defuzzification*; in this module fuzzy outputs from the inference module are converted to numerical crisp values; this is achieved by one of the several defuzzification algorithms; the most common is the Center of Gravity method, in which the crisp output value is calculated as the abscissa under the center of gravity of the output fuzzy set (see Figure 1.33).

In more complex technical applications of general function approximation (like in complex control systems, signal and image processing, etc.), two optional blocks are usually added to the fuzzy inference engine [Kos92, Kos96, Lee90]:

(0) *Preprocessor*, preceding the fuzzification module, performing various kinds of normalization, scaling, filtering, averaging, differentiation or integration of input data; and

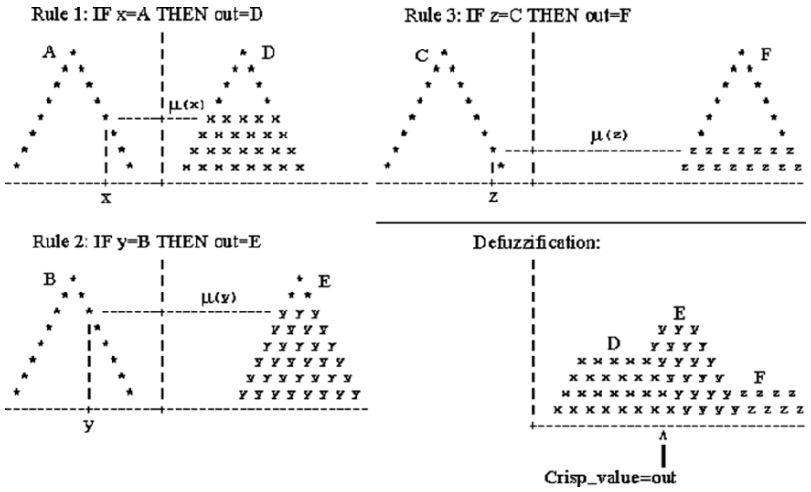

Fig. 1.33. Mamdani's Min–Max inference method and Center of Gravity defuzzification.

(4) *Postprocessor*, succeeding the defuzzification module, performing the analog operations on output data.

Common fuzzy systems have a simple feedforward mathematical structure, the so–called *Standard Additive Model* (SAM), which aids the spread of applications. Almost all applied fuzzy systems use some form of SAM, and some SAMs in turn resemble the ANN models (see [Kos99]).

In particular, an *additive fuzzy system* $F : \mathbb{R}^n \to \mathbb{R}^p$ stores m rules of the patch form $A_i \times B_i \subset \mathbb{R}^n \times \mathbb{R}^p$, or of the word form '**If** $X = A_i$ **Then** $Y = B_i$' and adds the 'fired' Then–parts $B_i^{'}(x)$ to give the output set $B(x)$, calculated as

$$B(x) = w_i B_i^{'}(x) = w_i \mu_i(x) B_i(x), \qquad (i = 1, \ldots, n), \tag{1.28}$$

for a scalar rule weight $w_i > 0$. The factored form $B_i^{'}(x) = \mu_i(x) B_i(x)$ makes the additive system (1.28) a SAM system. The fuzzy system F computes its output $F(x)$ by taking the centroid of the output set $B(x)$: $F(x) = \text{Centroid}(B(x))$. The *SAM theorem* then gives the centroid as a simple ratio,

$$F(x) = p_i(x) c_i, \qquad (i = 1, \ldots, n),$$

where the convex coefficients or discrete probability weights $p_i(x)$ depend on the input x through the ratios

$$p_i(x) = \frac{w_i \mu_i(x) V_i}{w_k \mu_k(x) V_k}, \qquad (i = 1, \ldots, n). \tag{1.29}$$

V_i is the finite positive volume (or area if $p = 1$ in the codomain space $\mathbb{R}^p$) [Kos99],

$$V_i = \int_{\mathbb{R}^p} b_i(y_1, \ldots, y_p) dy_1 ... dy_p > 0,$$

and c_i is the centroid of the Then–part set $B_i(x)$,

$$c_i = \frac{\int_{\mathbb{R}^p} y\, b_i(y_1, \ldots, y_p) dy_1 ... dy_p}{\int_{\mathbb{R}^p} b_i(y_1, \ldots, y_p) dy_1 ... dy_p}.$$

Fuzzy Logic Control

The most common and straightforward applications of fuzzy logic are in the domain of nonlinear control [Kos92, Kos96, Lee90, DSS96]. Fuzzy control is a nonlinear control method based on fuzzy logic. Just as fuzzy logic can be described simply as computing with words rather than numbers, fuzzy control can be described simply as control with sentences rather than differential equations.

A fuzzy controller is based on the fuzzy inference engine, which acts either in the feedforward or in the feedback path, or as a supervisor for the conventional PID controller.

A fuzzy controller can work either directly with fuzzified dynamical variables, like direction, angle, speed, or with their fuzzified errors and rates of change of errors. In the second case we have rules of the form:

1. IF error is Neg AND change in error is Neg THEN output is NB.
2. IF error is Neg AND change in error is $Zero$ THEN output is NM.

The collection of rules is called a rule base. The rules are in $IF-THEN$ format, and formally the $IF-$side is called the condition and the $THEN-$side is called the conclusion (more often, perhaps, the pair is called antecedent – consequent). The input value Neg is a linguistic term short for the word Negative, the output value NB stands for $Negative\_Big$ and NM for $Negative\_Medium$. The computer is able to execute the rules and compute a control signal depending on the measured inputs error and change in error.

The rule–base can be also presented in a convenient form of one or several rule matrices, the so–called $FAM-$matrices, where FAM is a shortcut for Kosko's *fuzzy associative memory* [Kos92, Kos96]. For example, a 9×9 graded FAM matrix can be defined in a symmetrical weighted form:

$$\text{FAM} = \begin{pmatrix} 0.6\text{S4} & 0.6\text{S4} & 0.7\text{S3} & \ldots & \text{CE} \\ 0.6\text{S4} & 0.7\text{S3} & 0.7\text{S3} & \ldots & 0.9\text{B1} \\ 0.7\text{S3} & 0.7\text{S3} & 0.8\text{S2} & \ldots & 0.9\text{B1} \\ \ldots & \ldots & \ldots & \ldots & 0.6\text{B4} \\ \text{CE} & 0.9\text{B1} & 0.9\text{B1} & \ldots & 0.6\text{B4} \end{pmatrix},$$

in which the vector of nine linguistic variables L^9 partitioning the *universes of discourse* of all three variables (with trapezoidal or Gaussian bell–shaped *membership functions*) has the form

$$L^9 = \{S4, S3, S2, S1, CE, B1, B2, B3, B4\}^T,$$

to be interpreted as: 'small 4', ... , 'small 1', 'center', 'big 1', ... , 'big 4'. For example, the left upper entry $(1,1)$ of the FAM matrix means: IF red is S4 and blue is S4, THEN result is 0.6S4; or, entry $(3,7)$ means: IF red is S2 and blue is B2, THEN result is center, etc.

Here we give three design examples for fuzzy controllers, the first one in detail, and the other two briefly.

Example: Mamdani Fuzzy Controller

The problem is to balance θ a pole of mass m and inertia moment I on a mobile platform of mass M that can be forced by F to move only (left/right) along $x-$axis (see Figure 1.34). This is quite an involved problem for conventional PID controller, based on differential equations of the pole and platform motion. Instead, we will apply fuzzy linguistic technique called *Mamdani inference* (see previous subsection).

Firstly, as a *fuzzification* part, we have to define (subjectively) what high speed, low speed etc. of the platform M is. This is done by specifying the membership functions for the fuzzy set partitions of the *platform speed* universe of discourse, using the following linguistic variables: (i) negative high

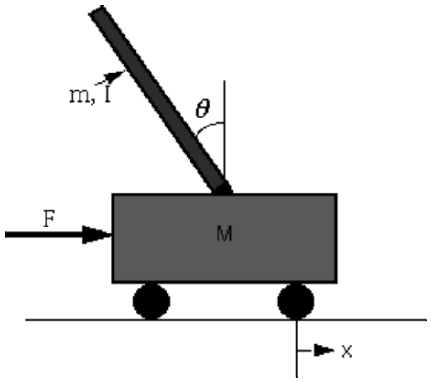

Fig. 1.34. Problem of balancing an inverted pendulum.

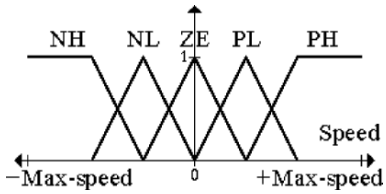

Fig. 1.35. Fuzzy membership functions for speed of the platform.

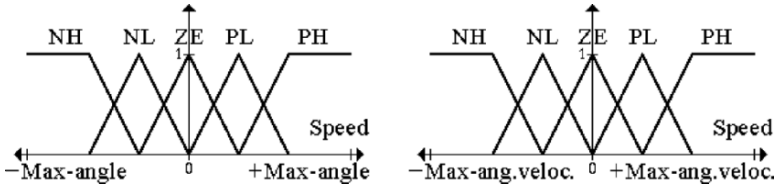

Fig. 1.36. Fuzzy membership functions for speed of the platform.

(NH), (ii) negative low (NL), (iii) zero (ZE), (iv) positive low (PL), and (v) positive high (PH) (see Figure 1.35).[152]

Also, we need to do the same for the angle θ between the platform and the pendulum and the angular velocity $\dot{\theta}$ of this angle (see Figure 1.36).

Secondly, as an *inference* part, we give several *fuzzy IF–THEN rules* that will tell us what to do in certain situations. Consider for example that the pole is in the upright position (angle θ is zero) and it does not move (angular velocity $\dot{\theta}$ is zero). Obviously this is the desired situation, and therefore we don't have to do anything (speed is zero). Let us consider also another case: the pole is in upright position as before but is in motion at *low* velocity in *positive*

[152] For simplicity, we assume that in the beginning the pole is in a nearly upright position so that an angle θ greater than, 45 degrees in any direction can never occur.

direction. Naturally we would have to compensate the pole's movement by moving the platform in the same direction at *low* speed.

So far we've made up two rules that can be put into a more formalized form like this:

IF angle is zero AND angular velocity is zero THEN speed shall be zero.

IF angle is zero AND angular velocity is positive low THEN speed shall be positive low.

We can summarize all applicable rules in the following FAM table (see previous subsection):

		Angle				
	Speed	NH	NL	ZE	PL	PH
V	NH			NH		
e	NL			NL	ZE	
l	ZE	NH	NL	ZE	PL	PH
o	PL		ZE	PL		
c	PH			PH		

Now, we are going to define two explicit values for angle and angular velocity to calculate with. Consider the situation given in Figure 1.37, and let us apply the following rule:

IF angle is zero AND angular velocity is zero THEN speed is zero

– to the values that we have previously selected (see Figure 1.38)

Only four rules yield a result (*rules fire*, see Figure 1.39), and we overlap them into one single result (see Figure 1.40).

Fan: the Temperature Control System

In this simple example, the input linguistic variable is:

$$temperature\_error = desired\_temperature - current\_temperature.$$

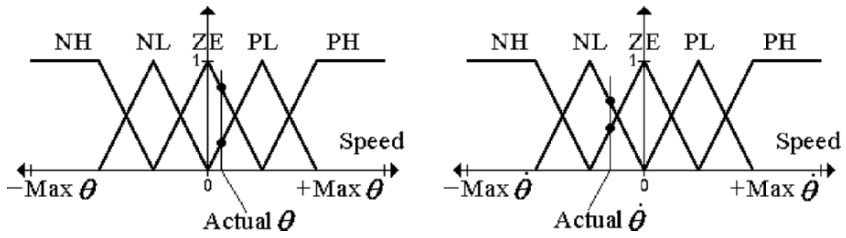

Fig. 1.37. Actual values for angle θ and angular velocity $\dot{\theta}$.

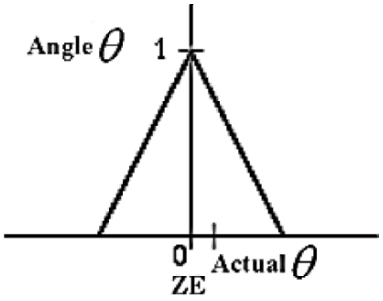

Fig. 1.38. Here is the linguistic variable *angle* θ where we zoom–in on the fuzzy set zero (ZE) and the actual angle.

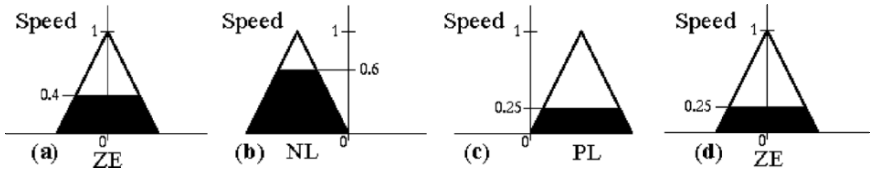

Fig. 1.39. Four fuzzy rules firing: (a) the result yielded by the rule: IF angle is zero AND angular velocity is zero THEN speed is zero; (b) the result yielded by the rule: IF angle is zero AND angular velocity is negative low THEN speed is negative low; (c) the result yielded by the rule: IF angle is positive low AND angular velocity is zero THEN speed is positive low; (d) the result yielded by the rule: IF angle is positive low AND angular velocity is negative low THEN speed is zero.

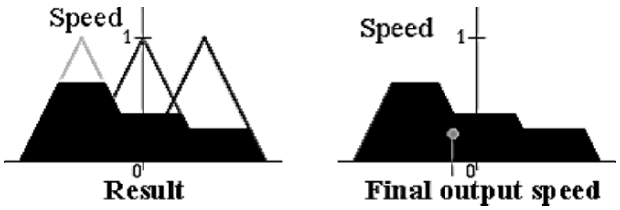

Fig. 1.40. Left: Overlapping single–rule results to yield the overall result. Right: The result of the fuzzy controller so far is a fuzzy set (of speed), so we have to choose one representative value as the final output; there are several heuristic *defuzzification* methods, one of them is to take the center of gravity of the fuzzy set. This is called *Mamdani fuzzy controller*.

The two output linguistic variables are:
hot_fan_speed, and $cool\_fan\_speed$. The universes of discourse, consisting of membership functions, i.e., overlapping triangular–trapezoidal shaped intervals, for all three variables are:

invar: $temperature\_error = \{Negative\_Big, Negative\_Medium, Negative\_Small, Zero, Positive\_Small, Positive\_Medium, Positive\_Big\}$, with the range $[-110, 110]$ degrees;

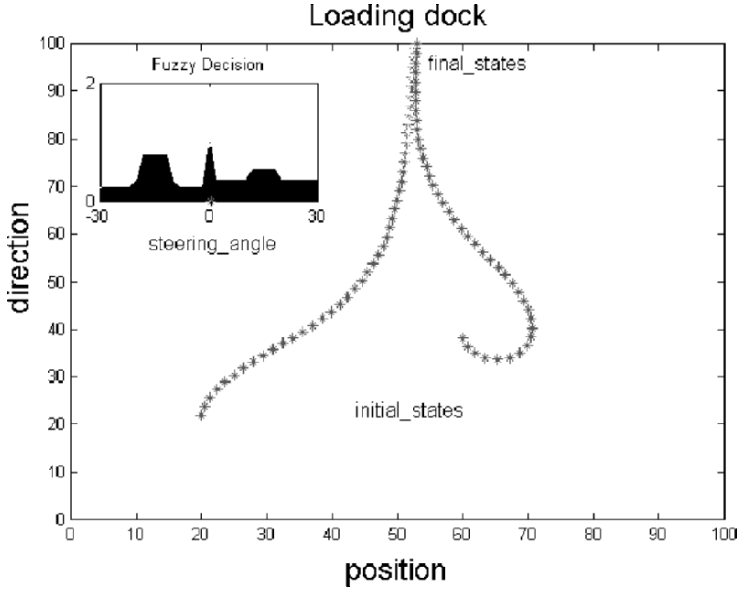

Fig. 1.41. Truck backer–upper steering control system.

outvars: hot_fan_speed and
$cool\_fan\_speed = \{zero, low, medium, high, very\_high\}$, with the range $[0, 100]$ rounds–per–meter.

Truck Backer–Upper Steering Control System

In this example there are two input linguistic variables: position and direction of the truck, and one output linguistic variable: steering angle (see Figure 1.41). The universes of discourse, partitioned by overlapping triangular–trapezoidal shaped intervals, are defined as:

invars: $position = \{NL, NS, ZR, PS, PL\}$, and
$direction = \{NL, NM, NS, ZR, PS, PM, PL\}$, where NL denotes Negative_Large, NM is Negative_Medium, NS is Negative_Small, etc.

outvar: $steering\_angle = \{NL, NM, NS, ZR, PS, PM, PL\}$.
The rule–base is given as:

IF direction is NL, AND position is NL, THEN steering angle is NL;
IF direction is NL, AND position is NS, THEN steering angle is NL;
IF direction is NL, AND position is ZE, THEN steering angle is PL;
IF direction is NL, AND position is PS, THEN steering angle is PL;
IF direction is NL, AND position is PL, THEN steering angle is PL;
IF direction is NM, AND position is NL, THEN steering angle is ZE;
. . . . . .
IF direction is PL AND position is PL, THEN steering angle is PL.

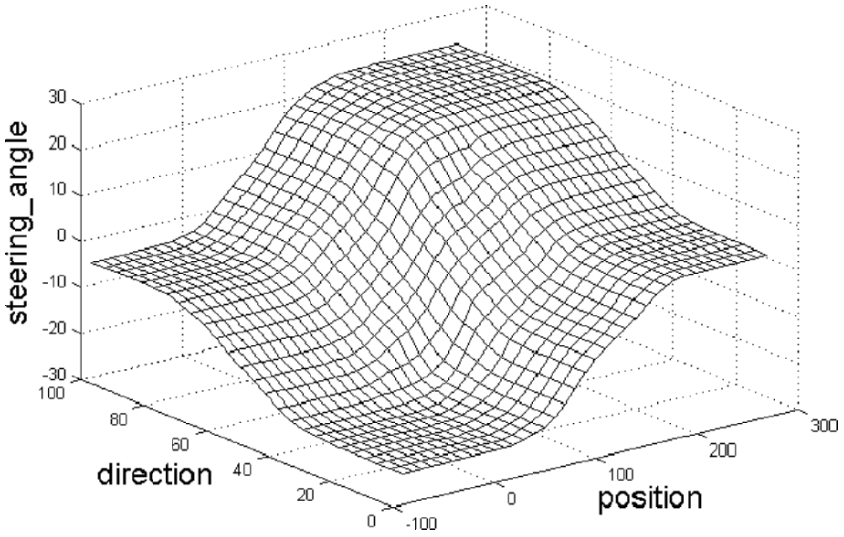

Fig. 1.42. Control surface for the truck backer–upper steering control system.

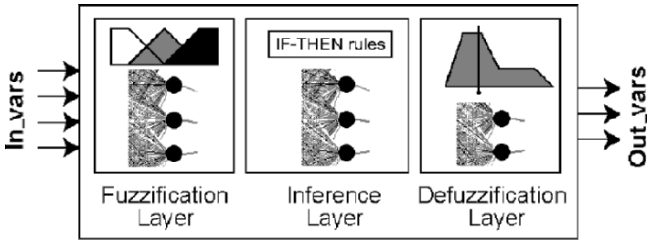

Fig. 1.43. Neuro–fuzzy inference engine.

The so–called *control surface* for the truck backer–upper steering control system is depicted in Figure 1.42.

To distinguish between more and less important rules in the knowledge base, we can put weights on them. Such weighted knowledge base can be then trained by means of artificial neural networks. In this way we get *hybrid neuro–fuzzy trainable expert systems.*

Another way of the hybrid neuro–fuzzy design is the fuzzy inference engine such that each module is performed by a layer of hidden artificial neurons, and ANN–learning capability is provided to enhance the system knowledge (see Figure 1.43).

Again, the fuzzy control of the BP learning (1.15–1.16) can be implemented as a set of heuristics in the form of fuzzy $IF-THEN$ rules, for the purpose of achieving a faster rate of convergence. The heuristics are driven by the behavior of the instantaneous sum of squared errors.

Finally, most *feedback fuzzy systems* are either discrete or continuous generalized SAMs [Kos99], given respectively by

$$x(k+1) = p_i(x(k))B_i(x(k)), \qquad \text{or} \qquad \dot{x}(t) = p_i(x(t))B_i(x(t)),$$

with coefficients p_i given by (1.29) above.

General Characteristics of Fuzzy Control

As demonstrated above, fuzzy logic offers several unique features that make it a particularly good choice for many control problems, among them [Lee90, DSS96]:

1. It is inherently robust since it does not require precise, noise–free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.
2. Since the fuzzy logic controller processes user–defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.
3. Fuzzy logic is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate–of–change parameters in order for it to be implemented. Any sensor data that provides some indication of a systems actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.
4. Because of the rule-based operation, any reasonable number of inputs can be processed (1–8 or more) and numerous outputs (1–4 or more) generated, although defining the rule–base quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller fuzzy logic controllers distributed on the system, each with more limited responsibilities.
5. Fuzzy logic can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

A *fuzzy logic controller* is usually designed using the following steps:

1. Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?
2. Determine the input and output relationships and choose a minimum number of variables for input to the fuzzy logic engine (typically error and rate–of–change of error).

3. Using the rule–based structure of fuzzy logic, break the control problem down into a series of *IF X AND Y THEN Z* rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the systems ability to minimize overshoot for a step inputs.
4. Create fuzzy logic membership functions that define the meaning (values) of Input/Output terms used in the rules.
5. Test the system, evaluate the results, tune the rules and membership functions, and re–test until satisfactory results are obtained.

Therefore, fuzzy logic does not require precise inputs, is inherently robust, and can process any reasonable number of inputs but system complexity increases rapidly with more inputs and outputs. Distributed processors would probably be easier to implement. Simple, plain–language rules of the form *IF X AND Y THEN Z* are used to describe the desired system response in terms of linguistic variables rather than mathematical formulas. The number of these is dependent on the number of inputs, outputs, and the designers control response goals. Obviously, for very complex systems, the rule–base can be enormous and this is actually the only drawback in applying fuzzy logic.

Evolving Fuzzy–Connectionist Systems

Recently, [Kas02] introduced a new type of fuzzy inference systems, denoted as dynamic evolving (see next subsection) neuro–fuzzy inference system (DENFIS), for adaptive online and off–line learning, and their application for dynamic time series prediction. *DENFIS system* evolves through incremental, hybrid (supervised/unsupervised), learning, and accommodates new input data, including new features, new classes, etc., through local element tuning. New fuzzy rules are created and updated during the operation of the system. At each time moment, the output of DENFIS is calculated through a fuzzy inference system based on $m-$most activated fuzzy rules which are dynamically chosen from a fuzzy rule set. Two approaches are proposed: (i) dynamic creation of a first–order Takagi–Sugeno–type (see, e.g., [Tan93]) fuzzy rule set for a DENFIS online model; and (ii) creation of a first–order Takagi–Sugeno–type fuzzy rule set, or an expanded high–order one, for a DENFIS offline model. A set of fuzzy rules can be inserted into DENFIS before or during its learning process. Fuzzy rules can also be extracted during or after the learning process. An evolving clustering method (ECM), which is employed in both online and off–line DENFIS models, is also introduced. It was demonstrated that DENFIS could effectively learn complex temporal sequences in an adaptive way and outperform some well–known, existing models.

Evolutionary Computation

In general, *evolutionary computation* (see [Fog98, ES03, BFM97]) is a CI–subfield involving *combinatorial optimization* problems.[153] It can be loosely recognized by the following criteria:

1. iterative progress, growth or development;
2. population based;
3. guided random search;
4. parallel processing; and
5. often biologically inspired.

This mostly involves the so–called *metaheuristic optimization algorithms*, such as *evolutionary algorithms* and *swarm intelligence*. In a lesser extent, evolutionary computation also involves *differential evolution*, *artificial life*, *artificial immune systems* and *learnable evolution model*.

Evolutionary Algorithms

In a narrow sense, evolutionary computation is represented by evolutionary algorithms (EAs), which are generic population–based *metaheuristic optimization algorithms* [Bac96]. The so–called *candidate solutions*[154] to the optimization problem play the role of individuals in a population, and the *cost function*[155] determines the environment within which the solutions 'live'. Evolution of the population then takes place after the repeated application of the above operators. Artificial evolution (AE) describes a process involving

[153] Recall that *combinatorial optimization* is a branch of optimization in applied mathematics and computer science, related to *operations research*, *algorithm theory* and *computational complexity theory*. Combinatorial optimization algorithms are often implemented in an efficient imperative programming language, in an expressive declarative programming language such as Prolog, or some compromise, perhaps a functional programming language such as Haskell, or a multi–paradigm language such as Lisp. A study of computational complexity theory helps to motivate combinatorial optimization. Combinatorial optimization algorithms are typically concerned with problems that are NP–hard. Such problems are not believed to be efficiently solvable in general. However, the various approximations of complexity theory suggest that some instances (e.g. 'small' instances) of these problems could be efficiently solved. This is indeed the case, and such instances often have important practical ramifications. The domain of combinatorial optimization is optimization problems where the set of *feasible solutions* is discrete or can be reduced to a discrete one, and the goal is to find the best possible solution.

[154] Recall that a *candidate solution* is a member of a set of possible solutions to a given problem. A candidate solution does not have to be a likely or reasonable solution to the problem. The space of all candidate solutions is called the *feasible region* or the feasible area.

[155] Recall that a generic optimization problem can be represented as:
Given: a function $f : A \to \mathbb{R}$ from some set A to the real numbers,

individual evolutionary algorithms; EAs are individual components that participate in an AE. EAs perform consistently well approximating solutions to all types of problems because they do not make any assumption about the underlying *fitness landscape*, evidenced by success in fields as diverse as engineering, art, biology, economics, genetics, operations research, robotics, social sciences, physics, and chemistry. Apart from their use as mathematical optimizers, EAs have also been used as an experimental framework within which to validate theories about biological evolution and natural selection, particularly through work in the field of artificial life. EAs involve biologically–inspired techniques implementing mechanisms such as:

1. *Reproduction*, which is the biological process by which new individual organisms are produced. Reproduction is a fundamental feature of all known life; each individual organism exists as the result of reproduction. The known methods of reproduction are broadly grouped into two main types: sexual and asexual. In asexual reproduction, an individual can reproduce without involvement with another individual of that species. The division of a bacterial cell into two daughter cells is an example of asexual reproduction. Asexual reproduction is not, however, limited to single–celled organisms. Most plants have the ability to reproduce asexually. On the other hand, sexual reproduction requires the involvement of

---

Sought: an element $x_0 \in A$ such that $f(x_0) \leq f(x)$ for all $x \in A$ ('minimization') or such that $f(x_0) \geq f(x)$ for all $x \in A$ ('maximization').

Typically, A is some subset of the *Euclidean space* $\mathbb{R}^n$, often specified by a set of constraints, equalities or inequalities that the members of A have to satisfy. The elements of A are called feasible solutions. The function f is called an *objective function*, or *cost function*. A feasible solution that minimizes (or maximizes, if that is the goal) the objective function is called an *optimal solution*. The domain A of f is called the *search space*, while the elements of A are called *candidate solutions* or *feasible solutions*.

Generally, when the feasible region or the objective function of the problem does not present *convexity*, there may be several local minima and maxima, where a local minimum x^* is defined as a point for which there exists some $\delta > 0$ so that for all x such that $\|\mathbf{x} - \mathbf{x}^*\| \leq \delta$, the expression $f(\mathbf{x}^*) \leq f(\mathbf{x})$ – holds; that is to say, on some region around x^* all of the function values are greater than or equal to the value at that point. Local maxima are defined similarly. For twice–differentiable functions, unconstrained problems can be solved by finding the points where the *gradient* of the objective function is zero (that is, the stationary points) and using the *Hessian matrix* to classify the type of each point. If the Hessian is positive definite, the point is a local minimum, if negative definite, a local maximum, and if indefinite it is some kind of saddle point. Constrained problems can often be transformed into unconstrained problems with the help of *Lagrange multipliers*. Note that a large number of algorithms proposed for solving non–convex problems, including the majority of commercially available solvers, are not capable of making a distinction between local optimal solutions and rigorous optimal solutions, and will treat the former as actual solutions to the original problem.

two individuals, typically one of each sex. Normal human reproduction is a common example of sexual reproduction. In general, more–complex organisms reproduce sexually while simpler, usually unicellular, organisms reproduce asexually.

2. *Mutation*, which is the biological change to the genetic material (usually DNA or RNA). Mutations can be caused by copying errors in the genetic material during cell division and by exposure to radiation, chemicals (mutagens), or viruses, or can occur deliberately under cellular control during processes such as meiosis or hypermutation. In multicellular organisms, mutations can be subdivided into germline mutations, which can be passed on to descendants, and somatic mutations. The somatic mutations cannot be transmitted to descendants in animals. Plants sometimes can transmit somatic mutations to their descendants asexually or sexually (in case when flower buds develop in somatically mutated part of plant). Mutations create variation in the gene pool, and then less favorable (or deleterious) mutations are removed from the gene pool by natural selection, while more favorable (beneficial or advantageous) ones tend to accumulate – this is evolution. Neutral mutations are defined as mutations whose effects do not influence the fitness of either the species or the individuals who make up the species. These can accumulate over time due to genetic drift. The overwhelming majority of mutations have no significant effect, since DNA repair is able to revert most changes before they become permanent mutations, and many organisms have mechanisms for eliminating otherwise permanently mutated somatic cells.
3. *Recombination*, which is the biological process of *genetic recombination* and *meiosis*, a genetic event that occurs during the formation of sperm and egg cells. It is also referred to as *crossover* or *phase change*.
4. *Natural selection*, which is the biological process by which individual organisms with favorable traits are more likely to survive and reproduce than those with unfavorable traits. Natural selection works on the whole individual, but only the heritable component of a trait will be passed on to the offspring, with the result that favorable, heritable traits become more common in the next generation. Given enough time, this passive process can result in adaptations and speciation. Natural selection is one of the cornerstones of modern biology. The term was introduced by Charles Darwin in his 1859 book 'The Origin of Species', by analogy with artificial selection, by which a farmer selects his breeding stock.
5. *Survival of the fittest*, a biological phrase, which is a shorthand for a concept relating to competition for survival or predominance. Originally applied by Herbert Spencer[156] in his 'Principles of Biology' of 1864, Spencer drew parallels to his ideas of economics with Charles Darwin's

[156] Herbert Spencer (27 April 1820 – 8 December 1903) was an English philosopher and prominent liberal political theorist. He is best known as the father of *Social Darwinism*, a school of thought that applied the evolutionist theory of survival of the fittest (a phrase coined by Spencer) to human societies. He also contributed to

theories of evolution by what Darwin termed natural selection. The phrase is a metaphor, not a scientific description; and it is not generally used by biologists, who almost exclusively prefer to use the phrase 'natural selection'.

Each evolutionary algorithm uses some mechanisms inspired by biological evolution: *reproduction*, *mutation*, *recombination*, *natural selection* and *survival of the fittest*. Candidate solutions to the optimization problem play the role of individuals in a population, and the cost function determines the environment within which the solutions 'live'. Evolution of the population then takes place after the repeated application of the above operators. The so–called *artificial evolution* (AE) describes a process involving individual evolutionary algorithms; EAs are individual components that participate in an AE.

Evolutionary algorithms perform consistently well approximating solutions to all types of problems because they do not make any assumption about the underlying fitness landscape, evidenced by success in fields as diverse as engineering, art, biology, economics, genetics, operations research, robotics, social sciences, physics, and chemistry. However, consider the no–free–lunch theorem.

Apart from their use as mathematical optimizers, EAs have also been used as an experimental framework within which to validate theories about biological evolution and natural selection, particularly through work in the field of artificial life. Techniques from evolutionary algorithms applied to the modelling of biological evolution are generally limited to explorations of microevolutionary processes, however some computer simulations, such as Tierra and Avida, attempt to model macroevolutionary dynamics.

In general, an evolutionary algorithm is based on three main statements:

1. It is a process that works at the chromosomic level. Each individual is codified as a set of chromosomes.
2. The process follows the Darwinian theory of evolution, say, the survival and reproduction of the fittest in a changing environment.
3. The evolutionary process takes place at the reproduction stage. It is in this stage when mutation and crossover occurs. As a result, the progeny chromosomes can differ from their parents ones.

Starting from a guess initial population, an evolutionary algorithm basically generates consecutive generations (offprints). These are formed by a set of chromosomes, or character (genes) chains, which represent possible solutions to the problem under consideration. At each algorithm step, a fitness function is applied to the whole set of chromosomes of the corresponding generation in order to check the goodness of the codified solution. Then, according

---

a wide range of subjects, including ethics, metaphysics, religion, politics, rhetoric, biology and psychology. Spencer is today widely criticized as a perfect example of scientism, while he had many followers and admirers in his time.

to their fitting capacity, couples of chromosomes, to which the crossover operator will be applied, are chosen. Also, at each step, a mutation operator is applied to a number of randomly chosen chromosomes.

The two most commonly used methods to randomly select the chromosomes are:

1. The *roulette wheel algorithm*. It consists in building a roulette, so that to each chromosome corresponds a circular sector proportional to its fitness.
2. The *tournament method*. After shuffling the population, their chromosomes are made to compete among them in groups of a given size (generally in pairs). The winners will be those chromosomes with highest fitness. If we consider a binary tournament, say the competition is between pairs, the population must be shuffled twice. This technique guarantees copies of the best individual among the parents of the next generation.

After this selection, we proceed with the sexual reproduction or crossing of the chosen individuals. In this stage, the survivors exchange chromosomic material and the resulting chromosomes will codify the individuals of the next generation. The forms of sexual reproduction most commonly used are:

(i) With one crossing point. This point is randomly chosen on the chain length, and all the chain portion between the crossing point and the chain end is exchanged.

(ii) With two crossing points. The portion to be exchanged is in between two randomly chosen points.

For the algorithm implementation, the crossover normally has an assigned percentage that determines the frequency of its occurrence. This means that not all of the chromosomes will exchange material but some of them will pass intact to the next generation. As a matter of fact, there is a technique, named elitism, in which the fittest individual along several generations does not cross with any of the other ones and keeps intact until an individual fitter than itself appears.

Besides the selection and crossover, there is another operation, mutation, that produces a change in one of the characters or genes of a randomly chosen chromosome. This operation allows to introduce new chromosomic material into the population. As for the crossover, the mutation is handled as a percentage that determines its occurrence frequency. This percentage is, generally, not greater than 5%, quite below the crossover percentage.

Once the selected chromosomes have been crossed and muted, we need some substitution method. Namely, we must choose, among those individuals, which ones will be substituted for the new progeny. Two main substitution ways are usually considered. In one of them, all modified parents are substituted for the generated new individuals. In this way an individual does never coexist with its parents. In the other one, only the worse fitted individuals of the whole population are substituted, thus allowing the coexistence among parents and progeny.

Since the answer to our problem is almost always unknown, we must establish some criterion to stop the algorithm. We can mention two such criteria [SRV04]:

(i) the algorithm is run along a maximum number of generations; and

(ii) the algorithm is ended when the population stabilization has been reached, i.e., when all, or most of, the individuals have the same fitness.

A limitation of EAs is their lack of a clear *genotype–phenotype distinction* [Bac96]. In nature, the fertilized egg cell undergoes a complex process known as embryogenesis to become a mature phenotype. This indirect encoding is believed to make the genetic search more robust (i.e., reduce the probability of fatal mutations), and also may improve the evolvability of the organism. Recent work in the field of artificial embryogeny, or artificial developmental systems, seeks to address these concerns.

Evolutionary algorithms usually comprise: *genetic algorithms, genetic programming, evolutionary programming, evolution strategy* and *learning classifier systems.*

Genetic Algorithms

The *genetic algorithm* (GA) is a search technique pioneered by John Holland[157] [Hol92] and used in computing to find true or approximate solutions to optimization and search problems (see [Gol89, Mit96, Vos99, Mic99]). GAs find application in computer science, engineering, economics, physics, mathematics and other fields. GAs are categorized as global search heuristics. GAs are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (mutated or recombined) to form a new population. The new population is then used

[157] John Henry Holland (February 2, 1929) is a pioneer in complex system and nonlinear science. He is known as the father of genetic algorithms. Holland is Professor of Psychology and Professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. He is also a member of The Center for the Study of Complex Systems (CSCS) at the University of Michigan, and a member of Board of Trustees and Science Board of the Santa Fe Institute. Holland is the author of a number of books about *complex adaptive systems* (CAS), including Hidden Order: How Adaptation Builds Complexity (1995), Emergence: From Chaos to Order (1998) and his ground–breaking book on genetic algorithms, 'Adaptation in Natural and Artificial Systems' (1975,1992). Holland also frequently lectures around the world on his own research, and on current research and open questions in CAS studies (see [Hol95, Hol95]).

in the next iteration of the algorithm. A typical GA requires two things to be defined: (i) a genetic representation of the solution domain, and (ii) a fitness function to evaluate the solution domain.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, that facilitates simple crossover operation. Variable length representations were also used, but crossover implementation is more complex in this case. The *fitness function*[158] is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the *knapsack problem*, we want to maximize the total value of objects that we can put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid, or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression; in these cases, *interactive genetic algorithms* are used. Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, and selection operators. Another way of looking at fitness functions is in terms of a *fitness landscape*,[159] which shows the fitness for each possible chromosome (see [Mit96]).

[158] A *fitness function* is a particular type of *objective function* that quantifies the optimality of a solution (that is, a chromosome) in a genetic algorithm so that particular chromosome may be ranked against all the other chromosomes. Optimal chromosomes, or at least chromosomes which are more optimal, are allowed to breed and mix their datasets by any of several techniques, producing a new generation that will (hopefully) be even better. An ideal fitness function correlates closely with the algorithm's goal, and yet may be computed quickly. Speed of execution is very important, as a typical genetic algorithm must be iterated many, many times in order to produce a useable result for a non–trivial problem. Definition of the fitness function is not straightforward in many cases and often is performed iteratively if the fittest solutions produced by GA are not what is desired. In some cases, it is very hard or impossible to come up even with a guess of what fitness function definition might be. Interactive genetic algorithms address this difficulty by out–sourcing evaluation to external agents (normally humans).

[159] Fitness landscapes or adaptive landscapes are used to visualize the relationship between genotypes (or phenotypes) and reproductive success. It is assumed that every genotype has a well defined replication rate (often referred to as fitness). This fitness is the 'height' of the landscape. Genotypes which are very similar are said to be 'close' to each other, while those that are very different are 'far'

It is well–known in biology that any organism can be represented by its *phenotype*, which virtually determines *what* exactly the object is in the real world, and its *genotype* containing all the information about the object at the chromosome set level. Each gene, that is the genotype's information element, is reflected in the phenotype. Thus, to be able to solve problems we have to represent every attribute of an object in a form suitable for use in genetic algorithms. All further operation of genetic algorithm is done on the genotype level, making the information about the object's internal structure redundant. This is why this algorithm is widely used to solve all sorts of problems.

In the most frequently used variant of genetic algorithm, an object's genotype is represented by bit strings. Each attribute of an object in the phenotype has a single corresponding gene in the genotype. The gene is represented by a bit string, usually of a fixed length, which represents the value of the attribute.

The simplest variant can be used to encode such attributes that is the bit value of the attribute. Then it will be quite easy to use a gene of certain length, sufficient to represent all possible values of such an attribute. Unfortunately this encoding method is not perfect. Its main disadvantage is that neighboring numbers differ in several bits' values. Thus, for example, such numbers as 7 and 8 in the bit representation have four different bits, which complicates the gene algorithm functioning and increases time necessary for its convergence. To avoid this problem another encoding method should be used, in which neighboring numbers have less differences, ideally differing in only one bit.

---

from each other. The two concepts of height and distance are sufficient to form the concept of a 'landscape'. The set of all possible genotypes, their degree of similarity, and their related fitness values is then called a fitness landscape. In evolutionary optimization problems, fitness landscapes are evaluations of a fitness function for all candidate solutions.

Apart from the field of evolutionary biology, the concept of a fitness landscape has also gained importance in evolutionary optimization methods, in which one tries to solve real–world engineering or logistics problems by imitating the dynamics of biological evolution. For example, a delivery truck with a number of destination addresses can take a large variety of different routes, but only very few will result in a short driving time. In order to use evolutionary optimization, one has to define for every possible solution s to the problem of interest (i.e., every possible route in the case of the delivery truck) how 'good' it is. This is done by introducing a scalar–valued function $f(s)$ (scalar valued means that $f(s)$ is a simple number, such as 0.3, while s can be a more complicated object, for example a list of destination addresses in the case of the delivery truck), which is called the fitness function or fitness landscape. A high $f(s)$ implies that s is a good solution. In the case of the delivery truck, $f(s)$ could be the number of deliveries per hour on route s. The best, or at least a very good, solution is then found in the following way. Initially, a population of random solutions is created. Then, the solutions are mutated and selected for those with higher fitness, until a satisfying solution has been found.

Binary coding			**Coding using the Gray code**		
Dec.code	Bin.value	Hex.value	Dec.code	Bin.value	Hex.value
0	0000	0h	0	0000	0h
1	0001	1h	1	0001	1h
2	0010	2h	3	0011	3h
3	0011	3h	2	0010	2h
4	0100	4h	6	0110	6h
5	0101	5h	7	0111	7h
6	0110	6h	5	0101	5h
7	0111	7h	4	0100	4h
8	1000	8h	12	1100	Ch
9	1001	9h	13	1101	Dh
10	1010	Ah	15	1111	Fh
11	1011	Bh	14	1110	Eh
12	1100	Ch	10	1010	Ah
13	1101	Dh	11	1011	Bh
14	1110	Eh	9	1001	9h
15	1111	Fh	8	1000	8h

Table 1.2. Correspondence between decimal codes and the Gray codes.

One of such codes is the Gray code, which is appropriate to be used with genetic algorithms. The table below shows the Gray code values:
Accordingly, when encoding an integer–valued attribute, we break it into quadruples and then convert each quadruple according to *Gray code*. Usually, there is no need to convert attribute values into gene values in practical use of GAs. In practice, inverse problem occurs, when it is necessary to find the attribute value from the corresponding gene value. Thus, the problem of decoding gene values, which have corresponding integer–valued attributes, is trivial. The simplest coding method, which first comes to mind, is to use bit representation. However, this variant is equally imperfect as in the case of integers. For this reason, the following sequence is used in practice:

1. All the interval of the attribute's allowed values is split into segments with adequate accuracy.
2. The value of the gene is accepted as an integer defining the interval number (using the Gray code).
3. The midpoint number of the interval is taken as the parameter value.

Let us consider a specific example of the sequence of operations described above: Assume that the attribute values are located in the interval $[0, 1]$. During the encoding the segment is split into 256 intervals. Thus we will need 8 bits to code their numbers. Let us suppose the number of the gene is $00100101bG$ (the capital letter 'G' stands for 'Gray code'). For a start we shall find the corresponding interval number using the following Gray code: $25hG \rightarrow 36h \rightarrow 54d$. Now let us see what interval corresponds to it ... Simple calculation gives us the interval: [0.20703125 , 0.2109375].

Then, the value of the parameter is $(0.20703125 + 0.2109375)/2 = 0.208984375$.

To encode nonnumeric data, we have to convert it into numbers. More detailed description can be found on our web site in the articles dedicated to the use of neural nets.

Thus, to find an object's *phenotype* (i.e., values of the attributes describing the object) we only have to know the values of the genes corresponding to these attributes, i.e., the object's *genotype*. The aggregate of the genes describing the object's genotype represents the *chromosome*. In some implementations it is called an individual. Thus, when implementing genetic algorithm, a chromosome is a bit string of a fixed length. Each segment of a string has its corresponding gene. Genes inside a chromosome can have equal or different lengths. Genes of equal length are used most often. Let us consider an example of a chromosome and interpretation of its value. Let us assume that the object has five attributes, each encoded by a gene 4 elements long. Then, the length of the chromosome is $5 \cdot 4 = 20$ bits:

0010	1010	1001	0100	1101

Now we can define the values of the attributes:

Attribute	Gene value	Binary value of the attribute	Decimal value of the attribute
Attribute 1	0010	0011	3
Attribute 2	1010	1100	12
Attribute 3	1001	1110	14
Attribute 4	0100	0111	7
Attribute 5	1101	1001	9

As it is known in the evolution theory, the way the parents' attributes are inherited by their offsprings is of high importance. In genetic algorithms an operator called *crossing* (also known as crossover or crossing over) is in charge of passing the attributes from parents to their offsprings. It works in the following way:

1. Two individuals are selected from the population to become parents;
2. A break point is determined (usually at random); and
3. The offspring is determined as concatenation of the first and the second parents' parts.

Let us see how this operator works:
Now, if we put the break after the third bit of the chromosome, then we have:

Chromosome_1:	0000000000
Chromosome_2:	1111111111

Chromosome_1:	0000000000	>>	000	1111111	Resulting_chromosome_1
Chromosome_2:	1111111111	>>	111	0000000	Resulting_chromosome_2

After that, one of the resulting chromosomes is taken as an offspring with the 0.5 probability.

The next genetic operator is intended for maintaining the diversity of individuals in the population. It is called *mutation*. When it is used on a chromosome, each bit in it gets inverted with a certain probability.

Besides, one more operator is used, called *inversion*. Applying it makes a chromosome break in two parts, which then trade places. This can be shown schematically as follows:

000	1111111	>>	1111111	000

Theoretically, these two genetic operators are enough to make the genetic algorithm work. However, in practice some additional operators are used, as well as modifications of these two operators. For instance, in addition to the single-point crossover (described above) there can be a multipoint one, when several break points (usually two) are formed. Besides, in some implementations of the algorithm the mutation operator performs the inversion of only one randomly selected bit of a chromosome.

Having found out how to interpret the values of the genes, we proceed to describing the genetic algorithm operation. Let us consider the flow chart of genetic algorithm operation in its classic variant.

1. Initialize the start time $t = 0$. At random fashion form the initial population consisting of k individuals: $B_0 = \{A_1, A_2, \ldots, A_\mathrm{k}\}$.
2. Calculate the *fitness* of every individual: $F_{\mathrm{A}i} = fit(A_\mathrm{i}), (i = 1 \ldots k)$, and of the population as a whole: $F_\mathrm{t} = fit(B_\mathrm{t})$. The value of this function determines how suitable for solving the problem the individual described by this chromosome is.
3. Select the individual A_c from the population: $A_\mathrm{c} = Get(B_\mathrm{t})$.
4. With a certain crossover probability P_c select the second individual from the population: $A_{\mathrm{c}1} = Get(B_\mathrm{t})$, and apply the crossover operator: $A_\mathrm{c} = Crossing(A_\mathrm{c}, A_{\mathrm{c}1})$.
5. With a certain mutation probability P_m apply the mutation operator: $A_\mathrm{c} = mutation(A_\mathrm{c})$.
6. With a certain inversion probability P_i apply the inversion operator: $A_\mathrm{c} = inversion(A_\mathrm{c})$.
7. Place the resulting chromosome in the new population: $insert(B_{\mathrm{t}+1}, A_\mathrm{c})$.

8. Repeat steps 3 to $7k$ times.
9. Increase the current epoch number $t = t + 1$.
10. If the stop condition is met, terminate the loop, else go to step 2.

Now let us examine in detail the individual steps of the algorithm. The steps 3 and 4 play the most important role in the successful operation of the algorithm when parent chromosomes are selected. Various alternatives are possible. The most frequently used *selection method* is called *roulette*. When using it, the probability of a chromosome selection is determined by its fitness, i.e.,

$$P_{\mathrm{Get(Ai)}} \sim Fit(A_{\mathrm{i}})/Fit(B_{\mathrm{t}}).$$

This method increases the probability of the attributes propagation that belong to the most adjusted individuals. Another frequently used method is the *tournament selection*. It means that several individuals (usually two) are selected in the population at random. The one wins which is more adjusted. Besides, in some implementations of the algorithm the so–called *elitism strategy* is used, which means that the best–adjusted individuals are guaranteed to enter the new population. Using the elitism method is usually helpful to accelerate the genetic algorithm convergence. The disadvantage of this strategy is increased probability of the algorithm getting in the local minimum.

Another important point is the algorithm stop criteria determination. Usually the highest limit of the algorithm functioning epochs is taken as such, or the algorithm is stopped upon stabilization of its convergence, normally measured by means of comparing the population's fitness on various epochs.

Genetic Programming

The *genetic programming* (GP) is an automated methodology inspired by biological evolution to find computer programs that best perform a user–defined task. It is therefore a particular machine learning technique that uses an evolutionary algorithm to optimize a population of computer programs according to a fitness landscape determined by a program's ability to perform a given computational task. The first experiments with GP were described in the book 'Genetic Programming' by John Koza (see [Koz92, Koz95, KBA99, KKS03]. Computer programs in GP can be written in a variety of programming languages. In the early (and traditional) implementations of GP, program instructions and data values were organized in tree–structures, thus favoring the use of languages that naturally embody such a structure (an important example pioneered by Koza is Lisp). Other forms of GP have been suggested and successfully implemented, such as the simpler linear representation which suits the more traditional imperative languages. The commercial GP software Discipulus, for example, uses linear genetic programming combined with machine code language to achieve better performance. Differently, the MicroGP uses an internal representation similar to linear genetic programming to generate programs that fully exploit the syntax of a given assembly language. GP is very computationally intensive and so in the 1990s it was mainly

used to solve relatively simple problems. However, more recently, thanks to various improvements in GP technology and to the well known exponential growth in CPU power, GP has started delivering a number of outstanding results. At the time of writing, nearly 40 human–competitive results have been gathered, in areas such as quantum computing, electronic design, game playing, sorting, searching and many more. These results include the replication or infringement of several post–year–2000 inventions, and the production of two patentable new inventions. Developing a theory for GP has been very difficult and so in the 1990s genetic programming was considered a sort of pariah amongst the various techniques of search. However, after a series of breakthroughs in the early 2000s, the theory of GP has had a formidable and rapid development. So much so that it has been possible to build exact probabilistic models of GP (schema theories and Markov chain models) and to show that GP is more general than, and in fact includes, GAs. On the other hand, techniques have now been applied to evolvable hardware as well as computer programs. Finally, the so–called *meta–GP* is the technique of evolving a GP–system using GP itself; critics have argued that it is theoretically impossible, but more research is needed.

Evolutionary Programming

The *evolutionary programming* (EP) was first used by Lawrence Fogel [FOW66] in 1960 in order to use simulated evolution as a learning process aiming to generate artificial intelligence. Fogel used finite state machines as predictors and evolved them. Currently evolutionary programming is a wide evolutionary computing dialect with no fixed structure, (representation), in contrast with the other three dialects. It is becoming harder to distinguish from evolutionary strategies. Its main variation operator is *mutation*; members of the population are viewed as part of a specific species rather than members of the same species therefore each parent generates an offspring, using a $(\mu + \mu)$ *survivor selection*.

Selection is the stage of a EP or GA in which individual genomes are chosen from a population for later breeding (recombination or crossover). There are several generic selection algorithms. One of the common ones is the so–called roulette wheel selection, which can be implemented as follows:

1. The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means multiplying the fitness value of each individual by a fixed number, so that the sum of all fitness values equals 1.
2. The population is sorted by descending fitness values.
3. Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of

the last individual should of course be 1 (otherwise something went wrong in the normalization step).
4. A random number R between 0 and 1 is chosen.
5. The selected individual is the first one whose accumulated normalized value is greater than R. There are other selection algorithms that do not consider all individuals for selection, but only those with a fitness value that is higher than a given (arbitrary) constant. Other algorithms select from a restricted pool where only a certain percentage of the individuals are allowed, based on *fitness value*.

Evolution Strategy

The *evolution strategy* (ES) is an optimization technique based on ideas of adaptation and evolution [Bey01, BS02]. ESs primarily use real–vector coding, and mutation, recombination, and environmental selection as its search operators. As common with EAs, the operators are applied in order:

1. mating selection,
2. recombination,
3. mutation,
4. fitness function evaluation, and
5. environmental selection.

Performing the loop one time is called a generation, and this is continued until a termination criterion is met. The first ES variants were not population based, but memorized only one search point (the parent) and one ((1+1)−ES) or more offspring ((1 + λ)−ES) at a time. Contemporary versions usually employ a population ((μ + λ)−ES) and are thus believed to be less prone to get stuck in local optima. Mutation is performed by adding a gaussian distributed random value simultaneously to each vector element. The step size or mutation strength (ie. the standard deviation of this distribution) is usually learned during the optimization. This process is called self–adaptation, and it should keep the evolutionary process within the *evolution window*.

It was observed in ES that during an evolutionary search the progress toward the fitness/objective function's optimum, generally, happens in a narrow band of mutation step size σ. That progress is called evolution window. So far, there is not an optimum tunning method for the mutation step size σ to keep the search inside the evolution window and how to fast achieve this window, although there are some investigations about that subject.

Learning Classifier Systems

The *learning classifier systems* (LCS) are machine learning systems with close links to reinforcement learning and genetic algorithms. First described by John Holland (see [Hol92, Hol95, Hol95]), an LCS consists of a population of binary rules on which a genetic algorithm altered and selected the best rules.

Instead of a using fitness function, rule utility is decided by a reinforcement learning technique. Learning classifier systems can be split into two types depending upon where the genetic algorithm acts. A Pittsburgh–type LCS has a population of separate rule sets, where the genetic algorithm recombines and reproduces the best of these rule sets. In a Michigan–style LCS there is only a single population and the algorithm's action focuses on selecting the best classifiers within that ruleset. Michigan–style LCSs have two main types of reinforcement learning, fitness sharing (ZCS) and accuracy–based (XCS). Initially the classifiers or rules were binary, but recent research has focused on improving this representation. This has been achieved by using populations of neural networks and other methods. Learning classifier systems are not well–defined mathematically and doing so remains an area of active research. Despite this, they have been successfully applied in many problem domains.

Swarm Intelligence

The *swarm intelligence* (SI) is based around the study of collective behavior in decentralized, self–organized systems (see, e.g., [Eng06]). The expression 'swarm intelligence' was introduced by Beni & Wang in 1989, in the context of *cellular automata*[160]. SI–systems are typically made up of a population of simple agents interacting locally with one another and with their environment. Although there is normally no centralized control structure dictating how individual agents should behave, local interactions between such agents often lead to the emergence of global behavior. Examples of systems like this can be found in nature, including ant colonies, bird flocking, animal herding, bacteria molding and fish schooling. Application of swarm principles to large numbers of robots is called as swarm robotics. SI–systems comprise:

1. The *ant colony optimization* (ACO), which is a *metaheuristic optimization algorithm* that can be used to find approximate solutions to difficult combinatorial optimization problems. In ACO artificial ants build solutions by moving on the problem graph and they, mimicking real ants, deposit artificial pheromone on the graph in such a way that future artificial ants can build better solutions. ACO has been successfully applied to an impressive number of optimization problems.

[160] Recall that a *cellular automaton* (plural: cellular automata, CA) is a discrete dynamical system invented by Stanislaw Ulam and John von Neumann. CA are studied in computability theory, mathematics, and theoretical biology. It consists of an infinite, regular grid of cells, each in one of a finite number of states. The grid can be in any finite number of dimensions. Time is also discrete, and the state of a cell at time t is a function of the states of a finite number of cells (called its neighborhood) at time $t-1$. These neighbors are a selection of cells relative to the specified cell, and do not change. Though the cell itself may be in its neighborhood, it is not usually considered a neighbor. Every cell has the same rule for updating, based on the values in this neighbourhood. Each time the rules are applied to the whole grid a new generation is created. See below for furhter details.

2. The *particle swarm optimization* (PSO), which is a global optimization algorithm for dealing with problems in which a best solution can be represented as a point or surface in an nD space. Hypotheses are plotted in this space and seeded with an initial velocity, as well as a communication channel between the particles. Particles then move through the solution space, and are evaluated according to some fitness criterion after each timestep. Over time, particles are accelerated towards those particles within their communication grouping which have better fitness values. The main advantage of such an approach over other global minimization strategies such as *simulated annealing* is that the large number of members that make up the particle swarm make the technique impressively resilient to the problem of local minima.
3. The *stochastic diffusion search* (SDS), which is an agent based probabilistic global search and optimization technique best suited to problems where the objective function can be decomposed into multiple independent partial–functions. Each agent maintains a hypothesis which is iteratively tested by evaluating a randomly selected partial objective function parameterised by the agent's current hypothesis. In the standard version of SDS such partial function evaluations are binary resulting in each agent becoming active or inactive. Information on hypotheses is diffused across the population via inter–agent communication. Unlike the stigmergetic communication used in ACO, in SDS agents communicate hypotheses via a $1-1$ communication strategy analogous to the tandem running procedure observed in some species of ant. A positive feedback mechanism ensures that, over time, a population of agents stabilise around the global–best solution. SDS is both an efficient and robust search and optimisation algorithm, which has been extensively mathematically described.

In a lesser extent, *evolutionary computation* also involves:

1. The *self–organization*,[161] comprising:
   a) The *self–organizing maps* (SOMs, or Kohonen[162] maps), which are a subtype of ANNs (see above), trained using unsupervised learning

[161] Recall that *self–organization* is a process in which the internal organization of a system, normally an open system, increases in complexity without being guided or managed by an outside source. Self–organizing systems usually display *emergent properties*. Self-organization usually relies on four basic ingredients: (i) positive feedback, (ii) negative feedback, (iii) balance of exploitation and exploration, and (iv) multiple interactions.

[162] Teuvo Kohonen, Dr. Ing (born July 11, 1934), is a Finnish academican and prominent researcher. He has made many contributions to the field of neural networks, including the Learning Vector Quantization algorithm, fundamental theories of distributed associative memory and optimal associative mappings, the learning subspace method and novel algorithms for symbol processing like

to produce low–dimensional representation of the training samples while preserving the topological properties of the input space; this makes SOMs especially good for visualizing high–dimensional data [Koh82, Koh88, Koh91]. SOM is a single layer feedforward network where the output neurons are arranged in low dimensional (usually 2D or 3D) grid. Each input is connected to all output neurons. Attached to every neuron there is a weight vector with the same dimensionality as the input vectors. The number of input dimensions is usually a lot higher than the output grid dimension. SOMs are mainly used for dimensionality reduction rather than expansion. The goal of SOM training is to associate different parts of the SOM lattice to respond similarly to certain input patterns. This is partly motivated by how visual, auditory or other sensory information is handled in separate parts of the cerebral cortex in the human brain. The weights of the neurons are initialized either to small random values or sampled evenly from the subspace spanned by the two largest principal component eigenvectors. The latter alternative will speed up the training significantly because the initial weights already give good approximation of SOM weights. The training utilizes competitive learning. Like most ANNs, SOM has two modes of operation:

i. During the training process a map is built, the neural network organises itself, using a competitive process. The network must be given a large number of input vectors, as much as possible representing the kind of vectors that are expected during the second phase (if any). Otherwise, all input vectors must be administered several times.
ii. During the mapping process a new input vector may quickly be given a location on the map, it is automatically classified or categorised. There will be one single winning neuron: the neuron whose weight vector lies closest to the input vector (this can be simply determined by calculating the Euclidean distance between input vector and weight vector).

b) The *growing neural gas* (GNG), which is a self–organized neural network proposed by B. Fritzke [Fri94]. It is based on the previously proposed *neural gas*, a biologically inspired adaptive algorithm, coined by Martinetz and Schulten in 1991, which sorts for the input signal according to how far away they are; a certain number of them are selected by distance in order, then the number of adaption units and strength are decreased according to a fixed schedule. On the other hand, GNG can add and delete nodes during algorithm execution.

---

redundant hash addressing. He has published several books and over 200 peer–reviewed papers. His most famous contribution is the self–organizing map (SOM) (also known as the Kohonen map, although Kohonen himself prefers SOM).

The growth mechanism is based on growing cell structures and competitive Hebbian learning.

c) The *competitive learning* (see, e.g., [Gro87]). In this area a large number of models exist which have a common goal to distribute a certain number of vectors in a possibly high–dimensional space. The distribution of these vectors reflects the probability distribution of the input signals which in general is not given explicitly but only through sample vectors. Two closely related concepts from computational geometry are the *Voronoi tessellation* and the *Delaunay triangulation* (see, e.g., [PS90]).

2. The *differential evolution* (DE), which grew out of K. Price's attempts to solve the *Chebyshev polynomial fitting problem* that had been posed to him by R. Storn. A breakthrough happened when Price came up with the idea of using vector differences for perturbing the vector population. Since this seminal idea a lively discussion between Price and Storn and endless ruminations and computer simulations on both parts yielded many substantial improvements which make DE the versatile and robust tool it is today. DE is a very simple population based, stochastic function minimizer which is very powerful at the same time. DE managed to finish 3rd at the First International Contest on Evolutionary Computation (Nagoya, 1996). DE turned out to be the best genetic type of algorithm for solving the real–valued test function suite of the 1st ICEO (the first two places were given to non–GA type algorithms which are not universally applicable but solved the test–problems faster than DE). The crucial idea behind DE is a scheme for generating trial parameter vectors. Basically, DE adds the weighted difference between two population vectors to a third vector. This way no separate probability distribution has to be used which makes the scheme completely self–organizing (see, e.g., [Lam02]).
3. The *artificial life* (*alife*), which is the study of life through the use of human–made analogs of living systems, evolving software that is more alive than a virus (see, e.g., [Lev92]). Theoretically, later it will become intelligent life. Computer scientist Christopher Langton coined the term in the late 1980s when he held the first Int. Conference on the Synthesis and Simulation of Living Systems (otherwise known as Artificial Life I) at the Los Alamos National Laboratory in 1987. Researchers of alife have focused on the 'bottom–up' nature of *emergent behaviors*. The alife field is characterized by the extensive use of computer programs and computer simulations which include evolutionary algorithms (EA), genetic algorithms (GA), genetic programming (GP), swarm intelligence (SI), ant colony optimization (ACO), artificial chemistries (AC), agent–based models, and cellular automata (CA). Often those techniques are seen as subfields of alife.The so–called *strong alife* position states that 'life is a process which can be abstracted away from any particular medium'.

Notably, Tom Ray declared that his program 'Tierra'[163] was not simulating life in a computer, but was synthesizing it. On the ther hand, the *weak alife* position denies the possibility of generating a 'living process' outside of a carbon–based chemical solution. Its researchers try instead to mimic life processes to understand the appearance of single phenomena. The usual way is through an agent based model, which usually gives a minimal possible solution. Closely related to alife is a *digital organism*, which is a self–replicating computer program that mutates and evolves. Digital organisms are used as a tool to study the dynamics of *Darwinian evolution*, and to test or verify specific hypotheses or mathematical models of evolution.

4. The *artificial immune system* (AIS), which is a type of optimisation algorithm inspired by the principles and processes of the vertebrate immune system (see [FPP86, Das99]). The algorithms typically exploit the immune system's characteristics of learning and memory to solve a problem. They are closely related to GAs. Processes simulated in AlS include pattern recognition, hypermutation and clonal selection for B cells,

[163] *Tierra* is a computer simulation developed by ecologist Thomas S. Ray in the early 1990s in which computer programs compete for central processing unit (CPU) time and access to main memory. The computer programs in Tierra are evolvable and can mutate, self–replicate and recombine. Tierra is a frequently cited example of an artificial life model; in the metaphor of the Tierra, the evolvable computer programs can be considered as digital organisms which compete for energy (CPU time) and resources (main memory). The basic Tierra model has been used to experimentally explore in silico the basic processes of evolutionary and ecological dynamics. Processes such as the dynamics of punctuated equilibrium, host–parasite co–evolution and density dependent natural selection are amenable to investigation within the Tierra framework. A notable difference to more conventional models of evolutionary computation, such as genetic algorithms is that there is no explicit, or exogenous fitness function built into the model. Often in such models there is the notion of a function being 'optimized'; in the case of Tierra, the fitness function is endogenous: there is simply survival and death. According to Ray and others this may allow for more 'open–ended' evolution, in which the dynamics of the feedback between evolutionary and ecological processes can itself change over time, although this promise has not been realized, like most other open–ended digital evolution systems, it eventually comes to a point where novelty ceases to be created, and the system at large begins either looping or evolving statically; some descendant systems like *Avida* try to avoid this pitfall. While the dynamics of Tierra are highly suggestive, the significance of the dynamics for real ecological and evolutionary behavior are still a subject of debate within the scientific community. Tierra is an abstract model, but any quantitative model is still subject to the same validation and verification techniques applied to more traditional mathematical models, and as such, has no special status. More detailed models in which more realistic dynamics of biological systems and organisms are incorporated is now an active research field.

negative selection of T cells, affinity maturation and immune network theory. In AIS, *antibody* and *antigen* representation is commonly implemented by strings of attributes. Attributes may be binary, integer or real–valued, although in principle any ordinal attribute could be used. Matching is done on the grounds of *Euclidean distance* $= \sum_{i=1}^{n}(x_i - y_i)^2$, *Manhattan distance*[164] or *Hamming distance*.[165] The so–called *clonal selection algorithms* are commonly used for *antibody hypermutation*. This allows the attribute string to be improved (as measured by a *fitness function*) using mutation alone.

5. The *learnable evolution model* (LEM), which is a novel, non–Darwinian methodology for evolutionary computation that employs machine learning

[164] The so–called *taxicab geometry*, considered by Hermann Minkowski in the 19th century, is a form of geometry in which the usual metric of Euclidean geometry is replaced by a new metric in which the distance between two points is the sum of the (absolute) differences of their coordinates. More formally, we can define the *Manhattan distance*, also known as the L^1-distance, between two points in an Euclidean space with fixed Cartesian coordinate system as the sum of the lengths of the projections of the line segment between the points onto the coordinate axes. Manhattan distance is also known as city block distance or taxi-cab distance. It is named so because it is the shortest distance a car would drive in a city laid out in square blocks, like Manhattan (discounting the facts that in Manhattan there are one–way and oblique streets and that real streets only exist at the edges of blocks, i.e., there is no 3.14th Avenue). Any route from a corner to another one that is 3 blocks East and 6 blocks North, will cover at least 9 blocks. All direct routes cover exactly 9. Taxicab geometry satisfies all of Hilbert's axioms except for the side–angle–side axiom, as one can generate two triangles with two sides and the angle between the same and have them not be congruent. A circle in taxicab geometry consists of those points that are a fixed Manhattan distance from the center. These circles are squares whose sides make a 45° angle with the coordinate axes.

In chess, the distance between squares on the chessboard for rooks is measured in Manhattan distance; kings and queens use Chebyshev distance, and bishops use the Manhattan distance (between squares of the same color) on the chessboard rotated 45 degrees, i.e., with its diagonals as coordinate axes. To reach from one square to another, only kings require the number of moves equal to the distance; rooks, queens and bishops require one or two moves (on an empty board, and assuming that the move is possible at all in the bishop's case).

[165] The Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different. Put another way, it measures the number of substitutions required to change one into the other, or the number of errors that transformed one string into the other. For example: (i) The Hamming distance between 1011101 and 1001001 is 2; (ii) The Hamming distance between 2143896 and 2233796 is 3; (iii) The Hamming distance between 'toned' and 'roses' is 3. The *Hamming weight* of a string is its Hamming distance from the zero string (string consisting of all zeros) of the same length. That is, it is the number of elements in the string which are not zero: for a binary string this is just the number of 1's, so for instance the Hamming weight of 11101 is 4.

to guide the generation of new individuals (candidate problem solutions) [WM06]. Unlike standard, Darwinian–type evolutionary computation methods that use random or semi–random operators for generating new individuals (such as mutations and/or recombinations), LEM employs hypothesis generation and instantiation operators. The hypothesis generation operator applies a machine learning program to induce descriptions that distinguish between high–fitness and low–fitness individuals in each consecutive population. Such descriptions delineate areas in the search space that most likely contain the desirable solutions. Subsequently the instantiation operator samples these areas to create new individuals.

Cellular Automata

It is common in nature to find systems whose overall behavior is extremely complex, yet whose fundamental component parts are each very simple. The complexity is generated by the cooperative effect of many simple identical components. Much has been discovered about the nature of the components in physical and biological systems; little is known about the mechanisms by which these components act together to give the overall complexity observed. According to Steve Wolfram [Wol02, Wol84], what is needed is a general mathematical theory to describe the nature and generation of complexity.

Cellular automata (CA) are examples of mathematical systems constructed from many identical components, each simple, but together capable of complex behavior. From their analysis one may, on the one hand, develop specific models for particular systems, and, on the other hand, hope to abstract general principles applicable to a wide variety of complex systems.

1D Cellular Automata

Recall that a 1D CA consists of a line of sites, with each site carrying a value 0 or 1 (or in general $0, \ldots, k-1$). The value α_i of the site at each position i is updated in discrete time steps according to an identical deterministic rule depending on a neighborhood of sites around it [Wol02, Wol84]:

$$\alpha_i^{t+1} = \varphi[\alpha_{i-r}^t, \alpha_{i-r+1}^t, \ldots, \alpha_{i+r}^t]. \tag{1.30}$$

Even with $k = 2$ and $r = 1$ or 2, the overall behavior of CA constructed in this simple way can be extremely complex.

Consider first the patterns generated by CA evolving from simple 'seeds' consisting of a few non–zero sites. Some local rules φ give rise to simple behavior; others produce complicated patterns. An extensive empirical study suggests that the patterns take on four qualitative forms (see Figure 1.44):

1. Disappears with time;
2. Evolves to a fixed finite size;
3. Grows indefinitely at a fixed speed; and
4. Grows and contracts irregularly.

Fig. 1.44. Classes of patterns generated by the evolution of CA from simple 'seeds'. Successive rows correspond to successive time steps in the CA evolution. Each site is updated at each time step according to equation (1.30) by CA rules that depend on the values of a neighborhood of sites at the previous time step. Sites with values 0 and 1 are represented by white and black squares, respectively. Despite the simplicity of their construction, patterns of some complexity are seen to be generated. The rules shown exemplify the four classes of behavior found. In the third case, a self–similar pattern is formed (adapted from [Wol02, Wol84]).

Patterns of type 3 are often found to be self–similar or scale invariant. Parts of such patterns, when magnified, are indistinguishable from the whole. The patterns are characterized by a *fractal dimension*, with the most common value $log_2 3 \simeq 1.59$. Many of the self–similar patterns seen in natural systems may in fact, be generated by CA evolution.

Different initial states with a particular CA rule yield patterns that differ in detail, but are similar in form and statistical properties. Different CA rules yield very different patterns. An empirical study, nevertheless, suggests that four qualitative classes may be identified, yielding four characteristic limiting forms:

1. Spatially homogeneous state;
2. Sequence of simple stable or periodic structures;
3. Chaotic aperiodic behavior; and
4. Complicated localized structures, some propagating.

All CA within each class, regardless of the details of their construction and evolution rules, exhibit qualitatively similar behavior. Such universality should make general results on these classes applicable to a wide variety of systems modelled by CA.

CA Applications

Mathematical models of natural systems are usually based on differential equations which describe the smooth variation of one parameter as a function of a few others. Cellular automata provide alternative and in some respects complementary models, describing the discrete evolution of many (identical) components. Models based on CA are typically most appropriate in highly nonlinear regimes of physical systems, and in chemical and biological systems

where discrete thresholds occur. Cellular automata are particularly suitable as models when *growth inhibition effects* are important [Wol02, Wol84].

As one example, CA provide global models for the growth of dendritic crystals (such as snowflakes). Starting from a simple seed, sites with values representing the solid phase are aggregated according to a 2D rule that accounts for the inhibition of growth near newly–aggregated sites, resulting in a fractal pattern of growth. Nonlinear chemical reaction–diffusion systems give another example: a simple CA rule with growth inhibition captures the essential features of the usual partial differential equations, and reproduces the spatial patterns seen. Turbulent fluids may also potentially be modelled as CA with local interactions between discrete vortices on lattice sites [Wol02, Wol84].

If probabilistic noise is added to the time evolution rule (1.30), then CA may be identified as generalized *Ising–spin* models. Phase transitions may occur if retains some deterministic components, or in more than one dimension.

Cellular automata may serve as suitable models for a wide variety of biological systems. In particular, they may suggest mechanisms for biological pattern formation. For example, the patterns of pigmentation found on many mollusc shells bear a striking resemblance to patterns generated by class 2 and 3 CA, and CA models for the growth of some pigmentation patterns have been constructed [Wol02, Wol84].

Two Approaches to CA Mathematics

Rather than describing specific applications of CA, here we concentrate on general mathematical features of their behavior. Two complementary approaches provide characterizations of the four classes of behavior [Wol02, Wol84].

In the first approach, CA are viewed as discrete dynamical systems (see, e.g., [GH83]), or discrete idealizations of partial differential equations. The set of possible (infinite) configurations of a CA forms a *Cantor set*. CA evolution may be viewed as a continuous mapping on this Cantor set. Quantities such as entropies, dimensions and Lyapunov exponents may then be considered for CA.

In the second approach, CA are instead considered as information–processing systems (see, e.g., [HU79]), or parallel–processing computers of simple construction. Information represented by the initial configuration is processed by the evolution of the CA. The results of this information processing may then be characterized in terms of the types of formal languages generated.[166]

[166] Note that the mechanisms for information processing in natural system appear to be much closer to those in CA than in conventional serial–processing computers: CA may, therefore, provide efficient media for practical simulations of many natural systems.

CA Entropies and Dimensions

Most CA rules have the important feature of irreversibility: several different configurations may evolve to a single configuration, and, with time, a contracting subset of all possible configurations appears. Starting from all possible initial configurations, the CA evolution may generate only special 'organized' configurations, and 'self–organization' may occur.

For class 1 CA, essentially all initial configurations evolve to a single final configuration, analogous to a limit point in a continuous dynamical system. Class 2 CA evolve to limit sets containing essentially only periodic configurations, analogous to limit cycles. Class 3 CA yield chaotic aperiodic limit sets, containing analogues of *strange attractors* [Wol02, Wol84].

Entropies and dimensions give a generalized measure of the density of the configurations generated by CA evolution. The (set) dimension or limiting (topological) entropy for a set of CA configurations is defined as (compare with [GH83]):

$$d^{(x)} = \lim_{X\to\infty} \frac{1}{X} \log_k N(X), \tag{1.31}$$

where $N(X)$ gives the number of distinct sequences of $X-$site values that appear. For the set of possible initial configurations, $d^{(x)} = 1$. For a limit set containing only a finite total number of configurations, $d^{(x)} = 0$. For most class 3 CA, $d^{(x)}$ decreases with time, giving, $0 < d^{(x)} < 1$, and suggesting that a fractal subset of all possible configurations occurs.

A dimension or limiting entropy $d^{(t)}$ corresponding to the time series of values of a single site may be defined in analogy with equation (1.31)[167] $d^{(t)} = 0$, for periodic sets of configurations.

Both $d^{(x)}$ and $d^{(t)}$ may be modified to account for the probabilities of configurations by defining

$$d_\mu^{(x)} = -\lim_{X\to\infty} \frac{1}{X} \sum_{i=1}^{k^\mu} p_i \log_k p_i, \tag{1.32}$$

and its $d^{(t)}-$analogue, where p_i are probabilities for possible length $X-$sequences. These measure dimensions may be used to delineate the large time behavior of the different classes of CA:[168]

1. $d_\mu^{(x)} = d_\mu^{(t)} = 0$;
2. $d_\mu^{(x)} > 0, d_\mu^{(t)} = 0$;
3. $d_\mu^{(x)} > 0, d_\mu^{(t)} > 0$.

[167] The analogue of equation (1.31) for a sufficiently wide patch of sites yields a topologically–invariant entropy for the CA mapping.

[168] Dimensions are usually undefined for class 4 CA.

CA Information Propagation

Cellular automata may also be characterized by the stability or predictability of their behavior under small perturbations in initial configurations, usually resulting from a change in a single initial site value (see Figure 1.45). Such perturbations have characteristic effects on the four classes of CA:

1. No change in final state;
2. Changes only in a finite region;
3. Changes over an ever-increasing region; and
4. Irregular changes.

In class 1 and 2 CA, information associated with site values in the initial state propagates only a finite distance; in class 3 CA, it propagates an infinite distance at a fixed speed, while in class 4 CA, it propagates irregularly but over an infinite range. The speed of information propagation is related to the Lyapunov exponent for the CA evolution, and measures the degree of sensitivity to initial conditions. It leads to different degrees of predictability for the outcome of CA evolution [Wol02, Wol84]:

1. Entirely predictable, independent of initial state;
2. Local behavior predictable from local initial state;
3. Behavior depends on an ever–increasing initial region; and
4. Behavior effectively unpredictable.

Information propagation is particularly simple for the special class of additive CA (whose local rule function φ is linear modulo k), in which patterns generated from arbitrary initial states may be obtained by superposition of patterns generated by evolution of simple initial states containing a single non–zero site. A rather complete algebraic analysis of such CA may be given. Most CA are not additive; however, with special initial configurations it is often possible for them to behave just like additive rules. Thus, for example, the evolution of an initial configuration consisting of a sequence of 00 and

Fig. 1.45. Evolution of small initial perturbations in CA, as shown by the difference (modulo two) between patterns generated from two disordered initial states differing in the value of a single site. The examples shown illustrate the four classes of behavior found. Information on changes in the initial state almost always propagates only a finite distance in the first two classes, but may propagate an arbitrary distance in the third and fourth classes (adapted from [Wol02, Wol84]).

01 diagrams under one rule may be identical to the evolution of the corresponding 'blocked' configuration consisting of 0 and 1 under another rule. In this way, one rule may simulate another under a blocking transformation (analogous to a renormalization group transformation). Evolution from an arbitrary initial state may be attracted to (or repelled from) the special set of configurations for which such a simulation occurs. Often several phases exist, corresponding to different blocking transformations: sometimes phase boundaries move at constant speed, and one phase rapidly takes over; in other cases, phase boundaries execute random walks, annihilating in pairs, and leading to a slow increase in the average domain size. Many rules appear to follow attractive simulation paths to additive rules, which correspond to fixed points of blocking transformations, and thus exhibit self similarity. The behavior of many rules at large times, and on large spatial scales, is therefore determined by the behavior of additive rules.

CA Thermodynamics

Decreases with time in the spatial entropies and dimensions of equations (1.31)–(1.32) signal irreversibility in CA evolution. Some CA rules are, however, reversible, so that each and every configuration has a unique predecessor in the evolution, and the spatial entropy and dimension of equations (1.31)–(1.32) remain constant with time.

Now, conventional *thermodynamics* gives a general description of systems whose microscopic evolution is reversible; it may, therefore, be applied to reversible CA. As usual, the 'fine–grained' entropy for sets (ensembles) of configurations, computed as in (1.32) with perfect knowledge of each site value, remains constant in time. The 'coarse–grained' entropy for configurations is, nevertheless, almost always non–decreasing with time, as required by the second law of thermodynamics. Coarse graining emulates the imprecision of practical measurements, and may be implemented by applying almost any contractive mapping to the configurations (a few iterations of an irreversible CA rule suffice). For example, coarse–grained entropy might be computed by applying (1.32) to every fifth site value. In an ensemble with low coarse–grained entropy, the values of every fifth site would be highly constrained, but arbitrary values for the intervening sites would be allowed. Then in the evolution of a class 3 or 4 CA the disorder of the intervening site values would 'mix' with the fifth–site values, and the coarse–grained entropy would tend towards its maximum value. Signs of self–organization in such systems must be sought in temporal correlations, often manifest in 'fluctuations' or meta–stable 'pockets' of order.

While all fundamental physical laws appear to be reversible, macroscopic systems often behave irreversibly, and are appropriately described by irreversible laws. Thus, for example, although the microscopic molecular dynamics of fluids is reversible, the relevant macroscopic velocity field obeys the irreversible *Navier–Stokes equations*. Conventional thermodynamics does not

apply to such intrinsically irreversible systems; new general principles must be found. Thus, for CA with irreversible evolution rules, coarse–grained entropy typically increases for a short time, but then decreases to follow the fine grained entropy. Measures of the structure generated by self–organization in the large time limit are usually affected very little by coarse graining.

CA and Formal Language Theory

Quantities such as entropy and dimension, suggested by information theory, give only rough characterizations of CA behavior. Computation theory suggests more complete descriptions of self–organization in CA (and other systems). Sets of CA configurations may be viewed as formal languages, consisting of sequences of symbols (site values) forming words according to definite grammatical rules.

The set of all possible initial configurations corresponds to a trivial formal language. The set of configurations obtained after any finite number of time steps are found to form a regular language. The words in a regular language correspond to the possible paths through a finite graph representing a finite state machine. It can be shown that a unique smallest finite graph reproduces any given regular language (see [HU79]). Examples of such graphs are shown in Figure 1.46. These graphs give complete specifications for sets of CA configurations (ignoring probabilities). The number of nodes in the smallest graph corresponding to a particular set of configurations may be defined as the 'regular language complexity' of the set. It specifies the size of the minimal description of the set in terms of regular languages. Larger correspond to more complicated sets.

The regular language complexity Ξ for sets generated by CA evolution almost always seems to be nondecreasing with time. Increasing Ξ signals increasing self–organization. Ξ may thus represent a fundamental property of self–organizing systems, complementary to entropy. It may, in principle, be extracted from experimental data [Wol02, Wol84].

Cellular automata that exhibit only class 1 and 2 behavior always appear to yields sets that correspond to regular languages in the large time limit. Class 3 and 4 behavior typically gives rise, however, to a rapid increase of Ξ with time, presumably leading to limiting sets not described by regular languages.

Formal languages are recognized or generated by idealized computers with a 'central processing unit' containing a fixed finite number of internal states, together with a 'memory'. Four types of formal languages are conventionally identified, corresponding to four types of computer:

1. Regular languages: no memory required.
2. Context–free languages: memory arranged as a last–in, first–out stack.
3. Context–sensitive languages: memory as large as input word required.
4. Unrestricted languages: arbitrarily large memory required (general Turing machine).

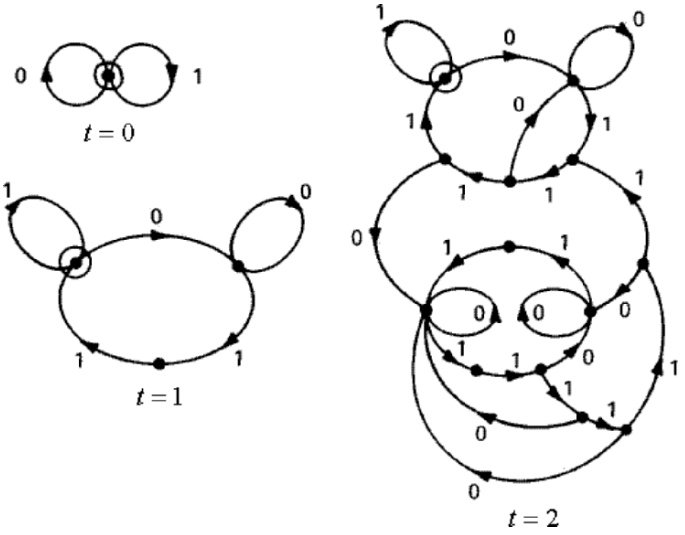

Fig. 1.46. Graphs representing the sets of configurations generated in the first few time steps of evolution according to a typical class 3 CA rule ($k = 2, r = 1$, rule number 126). Possible configurations correspond to possible paths through the graphs, beginning at the encircled node. At $t = 0$, all possible configurations are allowed. With time, a contracting subset of configurations are generated (e.g., after one time step no configuration containing the sequence of site value 101 can appear) At each time step, the complete set of possible configurations forms a regular formal language: the graph gives a minimal complete specification of it. The number of nodes in the graph gives a measure of the complexity Ξ of the set, viewed as a regular language. As for other class 3 CA, the complexity of the sets Ξ grows rapidly with time (modified and adapted from [Wol02, Wol84]).

Examples are known of CA whose limiting sets correspond to all four types of language. Arguments can be given that the limit sets for class 3 CA typically form context–sensitive languages, while those for class 4 CA correspond to unrestricted languages.[169]

CA and Computation Theory

While dynamical systems theory concepts suffice to define class 1, 2 and 3 CA, computation theory is apparently required for class 4 CA. Varied and complicated behavior, involving many different time scales is evident. Persistent structures are often generated. It seems that the structures supported by

[169] While a minimal specification for any regular language may always be found, there is no finite procedure to get a minimal form for more complicated formal languages; no generalization of the regular language complexity may thus be given.

this and other class 4 CA rule may be combined to implement arbitrary information processing operations. Class 4 CA would then be capable of universal computation: with particular initial states, their evolution could implement any finite algorithm. A few percent of CA rules with $k > 2$ or $r > 1$ are found to exhibit class 4 behavior: all these would then, in fact, be capable of arbitrarily complicated behavior. This capability precludes a smooth infinite size limit for entropy or other quantities: as the size of CA considered increases, more and more complicated phenomena may appear [Wol02, Wol84].

CA evolution may be viewed as a computation. Effective prediction of the outcome of CA evolution requires a short–cut that allows a more efficient computation than the evolution itself. For class 1 and 2 CA, such short cuts are clearly possible: simple computations suffice to predict their complete future. The computational capabilities of class 3 and 4 CA may, however, be sufficiently great that, in general, they allow no short-cuts. The only effective way to determine their evolution from a given initial state would then be by explicit observation or simulation: no finite formulae for their general behavior could be given.[170] Their infinite time limiting behavior could then not, in general, be determined by any finite computational process, and many of their limiting properties would be formally undecidable. Thus, for example, the 'halting problem' of determining whether a class 4 CA with a given finite initial configuration ever evolves to the null configuration would be undecidable. An explicit simulation could determine only whether halting occurred before some fixed time, and not whether it occurred after an arbitrarily long time.

For class 4 CA, the outcome of evolution from almost all initial configurations can probably be determined only by explicit simulation, while for class 3 CA this is the case for only a small fraction of initial states. Nevertheless, this possibility suggests that the occurrence of particular site value sequences in the infinite time limit is in general undecidable. The large time limit of the entropy for class 3 and 4 CA would then, in general, be non-computable: bounds on it could be given, but there could be no finite procedure to compute it to arbitrary precision.[171]

Undecidability and intractability are common in problems of mathematics and computation. They may well afflict all but the simplest CA. One may speculate that they are widespread in natural systems, perhaps occurring almost whenever nonlinearity is present. No simple formulae for the behavior of many natural systems could then be given; the consequences of their evolution could be found effectively only by direct simulation or observation.

For more details on CA, complexity and computation, see [Wol02].

---

[170] If class 4 CA are indeed capable of universal computation, then the variety of their possible behavior would preclude general prediction, and make explicit observation or simulation necessary.

[171] This would be the case if the limit sets for class 3 and 4 CA formed at least context–sensitive languages.

Adaptive Business Intelligence

Recall that businesses and government agencies are mostly interested in two fundamental things [Mic06]: (i) knowing what will happen next (prediction), and (ii) making the best decision under risk and uncertainty (optimization) (see Figure 1.47). Therefore, from CI–perspective, the goal is to provide CI–based solutions for modelling, simulation, and optimization to address these two fundamental needs.

Information technology applications that support decision–making processes and problem–solving activities have proliferated and evolved over the past few decades. In the 1970s, these applications were simple and based on spreadsheet software. During the 1980s, decision-support systems incorporated optimization models, which originated in the operations research and management science communities. In the 1990s, these systems were further enhanced with components from artificial intelligence and statistics [MSM05]. This evolution led to many different types of *decision–support systems* with somewhat confusing names, including *management information systems*, *intelligent information systems*, *expert systems*, *management–support systems*, and *knowledge–based systems*. Because businesses realized that data was a precious asset, they often based these 1intelligent' systems on data warehousing and online analytical processing technologies. They gathered and stored a lot of data, assuming valuable assets were implicitly coded in it. Raw data, however, is rarely beneficial. Its value depends on a user's ability to extract knowledge that is useful for decision support. Thousands of 'business intelligence' companies thus emerged to provide such services. After analyzing a corporation's operational data, for example, these companies might return intelligence (in the form of tables, graphs, charts, and so on) stating that, say,

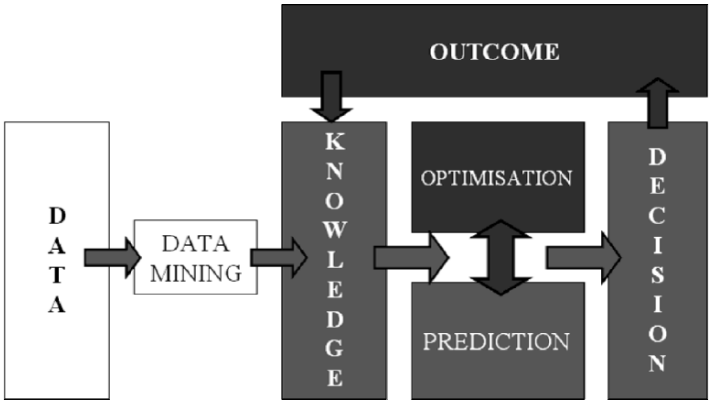

Fig. 1.47. Adaptive business intelligence: the diagram shows the flow from data acquisition to recommended action, including an adaptive feedback loop (adapted from [Mic06]).

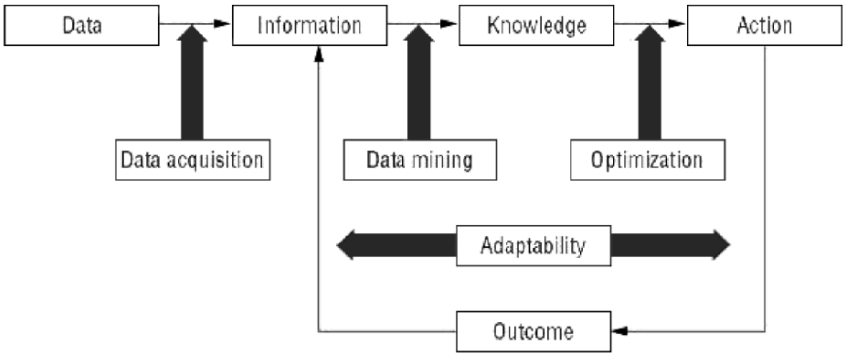

Fig. 1.48. Adaptive business intelligence: the diagram shows the flow from data acquisition to recommended action, including an adaptive feedback loop (adapted from [MSM05]).

57 percent of the corporation's customers are between 40 and 50, or product Q sells much better in Florida than in Georgia.

Many businesses have realized, however, that the return on investment for pure 'business intelligence' is much smaller than initially thought. The 'discovery' that 57 percent of our customers are between 40 and 50 doesn't directly lead to decisions that increase profit or market share. Moreover, we live in a dynamic environment where everything is in flux. Interest rates change, new fraud patterns emerge, weather conditions vary, the stock markets rise and fall, new regulations and policies surface, and so on. These economic and environmental changes render some data obsolete and make other data—which might have been useless just six weeks ago—suddenly meaningful.

Michalewicz *et al.* developed a software system (see Figure 1.48) to address these complexities and implemented it on a real distribution problem for a large car manufacturer. The system detects data trends in a dynamic environment, incorporates optimization modules to recommend a near–optimum decision, and includes self–learning modules to improve future recommendations. As Figure 1.48 shows, such a system lets enterprizes monitor business trends, evolve and adapt quickly as situations change, and make intelligent decisions based on uncertain and incomplete information. This intelligent system combines three modules: prediction, optimization and adaptation.

Research Issues in Dynamic Optimization

Most data–mining and optimization algorithms assume static data and a static objective. Typically, they search for a snapshot of 'knowledge' and a near-optimum solution with respect to some fixed measure (or set of measures), such as profit maximization or minimization of task–completion time. However, real–world applications operate in dynamic environments, where it's often necessary to modify the current solution due to changes in the problem

setting, such as machine breakdown or employee illness; or the environment, such as consumer trends or changes in weather patterns or economic indicators. It's therefore important to investigate adaptive algorithms that don't require restart every time a change is recorded. In many commercial situations, such restarts are not an option.

Evolutionary Techniques

An obvious starting point here is evolutionary computation techniques [MF04], which are optimization algorithms inspired by the continuously changing natural environment. However, it is important to investigate which evolutionary algorithm extensions are actually useful in business scenarios. Unfortunately, most current approaches ignore dynamics and assume that re–optimization should occur at regular intervals. However, significant benefits can be realized when researchers explicitly address dynamism.

Many researchers have proposed various benchmarks for studying optimization in dynamic environments. Among the proposals are the moving peaks benchmark, the dynamic knapsack problem, dynamic bit–matching, scheduling with new jobs arriving over time, and the greenhouse control problem. Researchers have also proposed various measures, including off–line error, percentage of covered peaks, and diversity. Among the partial conclusions reached in this research [Bra01]:

- standard evolutionary algorithms get stuck on a single peak;
- diversity preservation slows down the convergence;
- random immigrants introduce high diversity from the beginning, but offer limited benefits;
- memory without diversity preservation is counterproductive; and
- nonadaptive memory suffers significantly if peaks move.

However, several essential points are seemingly missing in the key research on optimization in dynamic environments. Most researchers emphasize an ultimate goal of approximating real–world environments, but they fail to address several key issues for successful adaptive–system development. The following issues, which constitute the conceptual research framework, are essential for creating a methodology for building *intelligent systems* [MSM05].

Non–Stationary Constraints

Here, the task is to optimize a non–stationary *objective function* $f(x,t)$, subject to non–stationary constraints, $ci(x,t) \leq 0, (i = 1, 2, \ldots, k)$. This approach was applied successfully in the context of a collision situation at sea [SM00]. By accounting for particular maneuvering–region boundaries, along with information on navigation obstacles and other moving ships, the authors reduced the *collision–avoidance problem* to a *dynamic optimization task* with static and dynamic constraints. The proposed algorithm computed a safe and optimum ship path in both static and dynamic environments.

Prediction Component

Environmental changes are seldom random. In a typical real–world scenario, where constraints change over time, it's possible to calculate some failure probabilities by analyzing past data, and thus predict a possible environmental change. The above mentioned work on collisions at sea [SM00] offers a good example here as well. The authors based a ship's safe trajectory in a collision situation on predicted speeds and the other ships' directions. Studying dynamic environments where change is somewhat predictable is important, but so far, little work exists along these lines.

Parameter Adaptation

In nonstationary environments, researchers must study parameter control, particularly when the adaptive system includes predictive methods [EHM99].

Solution Robustness

Research into robustness concentrates on questions such as: What constitutes flexibility in the specific context? How can we integrate a flexibility goal into the algorithm? To answer these questions, we must take into account a predictive model (for environmental changes) and the prediction's estimated error. This has yet to occur [MSM05]. Many researchers have recognized the importance of solution robustness [Bra01]. Existing approaches vary, from techniques to 'disturb' individuals in the population to those using search history. Some researchers have considered an aspect of robustness, sometimes called flexibility, in which the problem requires sequential decision–making under an uncertain future, and the decision influences the system's future state. In such situations, the decision–making process should anticipate future needs. That is, rather than focusing exclusively on the primary objective function, it should try to move the system into a flexible state.