A Structural Approach to Infer Recurrent Relations in Data

Pietro Cottone, Salvatore Gaglio and Marco Ortolani

Abstract Extracting knowledge from a great amount of collected data has been a key problem in Artificial Intelligence during the last decades. In this context, the word "knowledge" refers to the non trivial new relations not easily deducible from the observation of the data. Several approaches have been used to accomplish this task, ranging from statistical to structural methods, often heavily dependent on the particular problem of interest. In this work we propose a system for knowledge extraction that exploits the power of an ontology approach. Ontology is used to describe, organise and discover new knowledge. To show the effectiveness of our system in extracting and generalising the knowledge embedded in data, we have built a system able to pick up some strategies in the solution of complex puzzle game.

1 Introduction

During the last decades the ever-decreasing cost of wireless sensors and actuators has allowed an increasing diffusion of pervasive networks to monitor and control every kind of environment. In this context a new paradigm was conceived, namely the *Internet of Things* (*IoT*). The aim of this paradigm is to allow to a large set of different appliances in the environment to interact with each other and cooperate to get common goals [1], through an Internet-like structure and a unique addressing scheme. The availability of such technologies has pushed for the creation of better Decision Support Systems (DSS), able to take advantage of the richness present in

P. Cottone · S. Gaglio · M. Ortolani (✉)
DICGIM, Viale delle Scienze, University of Palermo, Edificio 6, 90128 Palermo, Italy
e-mail: marco.ortolani@unipa.it

S. Gaglio
e-mail: salvatore.gaglio@unipa.it

P. Cottone
e-mail: pietro.cottone@unipa.it

S. Gaglio and G. Lo Re (eds.), *Advances onto the Internet of Things*,
Advances in Intelligent Systems and Computing 260, DOI: 10.1007/978-3-319-03992-3_8,

data. *Ambient Intelligence* (*AmI*) is an emerging framework in this scenario, whose aim is to make the environment aware of the user presence and thus supporting them to perform every-day activities. Meeting the goals of *AmI* means understanding what is happening in the monitored area, in order to plan a set of actions on the actuators so as to modify the environment conditions according to user's desires. An *AmI* system, therefore, must deal with high level concepts, expressed through simple sensor readings. It analyses a great amount of rough data (i.e, sensor measures), coming from the environment, and summarises them in a high-level representation, through some concepts and their relations. In other words, the *AmI* system must be able to extract knowledge from a great amount of sensory data, giving an explanation of data itself. According to some proposals, an *AmI* system acts like an agent [2, 3], but an agent needs a model of the environment to operate; therefore, the designer of an *AmI* system has to embed some a-priori knowledge into the system, in order to code this model. Obviously, this task can take advantage of the analysis of such a great amount of sensory data, but comprehension of the data and the following translation into usable knowledge is not an easy task. In fact, "measuring" does not directly translate into "understanding", so sensory data provided from pervasive networks can not be easily turned in a corresponding new knowledge. Moreover, extracted knowledge would be easily generalizable: similar problems have similar solutions, so knowledge can be summarily defined as the common structure shared among similar problem solution, to construct a general model of the environment. Thus, constructing a new model for every instance of similar problems may be redundant. Nowadays, this kind of knowledge is only saved in the experience of the designer and there is no automatic system to extract it or to aid the designer in this task. However, new challenges coming from *IoT* or *AmI* call for the creation of system able to learn from experience, that is capable of capturing the hidden structure of the data, in terms of relations between its key components.

All this problems are related to knowledge *representation*, *management* and *reuse*, i.e. they are an *ontology* problem [4]. The ontology notion comes from philosophy, where it refers to the metaphysical study of the nature of being and existence. In computer science, and more specifically in the field of knowledge engineering, ontologies are used for modelling concepts and relationships on some expertise domain. Thus, building an ontology of the most relevant entities of a scenario in a semi-automatic fashion is a key problem in emerging technologies and a cutting-edge challenge of Artificial Intelligence. It involves different research areas (e.g., data mining, planning, etc), but the most interesting formulation, according to our vision, is the one arising from machine learning. In fact, this problem can be formulated as the creation of a system able to construct and recognise likely explanation of a great amount of data, unveiling their hidden structure. All the approaches used nowadays (statistical, syntactic, logic, etc) are showing their limits and inappropriateness. In fact, it is very difficult (and maybe impossible) to use only one of these approaches to manage very high level concept, to model the living world and make sense of it. Moreover, the development of domain ontology has been a task entirely based on human intervention. But new applications in *IoT* require the management of such a large number of concepts that is impossible to be performed by a human alone [5].

So, the availability of semi-automated (or, less likely, full-automated) ontology systems for the management and discovery of new knowledge is a key point in the development of actually useful DSS. There are several approaches to create semi-automated ontology learning systems, but none of them has been applied to the field of sensory data. Most of them have been used on semantic web data or huge text corpora [5, 6].

We claim that the expressive power of structural approaches is the key to handle the complexity of acquiring knowledge from unstructured data and related to every-day situation. The idea of the proposed work is to describe a general framework to deal with this problem, using as example application the problem of finding strategies. Given a problem description, whose solution is obviously unknown, and a set of solution examples (our rough data), we aim to abstract general guidelines about problem solution. This implies highlighting the common characteristic of solution and obtaining a general description of the solution itself, in terms of its key components. In particular, we address the problem of finding a good heuristic for the well-known slide tile puzzle, using structural information.

The rest of the paper is organised as follows. Section 2 summarizes some of the approaches presented in literature, with regards to knowledge extraction tools. Section 3 describes the problem of Ontology Learning and our proposal to deal with unstructured data corpora, such as sensory data. Section 4 proposes a testbed to evaluate our approach. Finally, in Sect. 5, our conclusions are reported.

2 Related Work

The need for coupling semantics with a sequence of sensor readings is well-known in literature. In fact, inferring knowledge from data is an open issue in Computer Science, and in particular in the area of data mining [7]. In this context, defining what can be deemed as *interesting* knowledge is a hard problem, because it implies to find what can be interpreted as an important information. Historically, a first debate on the most profitable way to extract useful information (i.e., knowledge) from a data collection was opened by John Tukey [8]. In the seventies, he proposed the *Exploratory Data Analisys* (*EDA*), as opposite to the *Confirmatory Data Analysis* (*CDA*) or *Statistical Hypothesis Testing*, that was the standard approach in those years. In the *EDA* approach, data are analysed with different techniques to summarize their characteristics. Unlike *CDA*, Tukey suggests to let hypotheses emerge from data themselves, rather than using data only to test a-priori hypotheses. The *Exploratory Data Analisys* is just an approach, not a set of techniques, i.e. a suggestion about how data analysis should be carried out and what its goals should be. Most of the techniques inspired to *EDA* use a graphical approach, because it represents a very powerful instrument to reveal the structure of the data to the analyst, offering new and often unexpected insights. In other words, it empowers the analyst's natural pattern-recognition capabilities. Therefore, it was the seminal work of modern approaches to data mining and pattern recognition.

One of the contemporary and independent developed research carried out on the track of *EDA* is the so-called *GUHA* (*General Unary Hypotheses Automaton*) principle [9]. The aim is to describe all assertions which may be hypotheses, verify each of such assertions and found the “interesting” ones, based on collected data. These systems generate systematically all interesting hypotheses with respect to the given data (hypotheses describing relations among properties of objects) via a standard computer system, and therefore represent a first attempts to formalise an automatic inductive approach. Logic is used to formulate hypotheses, coded as association of properties. Each object is represented by a row in a rectangular matrix, whose column are properties of the object. Analysing this data structure is possible to discover dependencies between different properties. The whole process is composed by three steps: *preprocessing*, *kernel* and *post-processing*. In the first step, matrix is arranged in a form suitable for a quick hypothesis generation. In the *kernel* phase, hypotheses are generated and evaluated, while in the last step hypotheses are analysed in order to interpret them.

It is crucial to note that the problem of letting structure and explanation emerge from data itself and not from a-priori hypotheses was central since the beginning of data analysis history, and has gained more relevance over the years, due to the ever-increasing size and heterogeneity that have characterised the data to analyse. Nowadays, the collected sensory data make it impossible to promote a-priori hypotheses to describe events of interest. The discussion between *EDA* and *CDA* approaches has renewed in the machine learning. In fact, two different approaches have grown in importance: *inductive* and *deductive* learning. This distinction reflects the differences and goals already underlined by Tukey, with a special focus of attention to the learning matter. The inductive approaches state the learning problem as finding a hypothesis that agrees with the examples, preferring the most simple one. It includes a variety of algorithms, such as instance-based learning, Support Vector Machines, Naïve Bayes, Artificial Neural Network, etc. Each of these approaches stresses different aspects of learning problem, but they relieve the analyst and designer from formulating an a-priori hypothesis about data. On the other hand, their responses are not useful to increase the knowledge regarding a particular problem because they can be considered as a black-box that can be applied on unseen data, but the model of the data they use is not human interpretable.

The deductive learning approaches constitute the other class of machine learning algorithms. For example, a method to infer general concepts from examples is known as *Explanation-Based Generalisation* (*EBG*) [10]. This deductive approach explains why a training example is a member of the concept being learned. This approach relies on four main components: a goal concept, training example, domain theory and operational criterion. Explanations are represented by Horn-clause inference rules arranged in proof trees. The goal concept is described through high-level properties that are not directly found in the example. Training example is a representation of a specific example in terms of lower level features. The domain theory is made up of a set of inference rules and axioms about the domain of interest. Domain theory is used to demonstrate the validity of the example. The operational criterion indicates how a concept must be expressed to be recognised. The aim of the system is just to generalise concepts from examples. A slightly different approach is that proposed by [11]. In this case the system is not only able to generalise a concept, but to check where a generalisation fails for a particular example, so that the system can refine it. Therefore it is possible not only to infer a general concept, but also to check whether an example is coherent with that generalisation, or why it is not; in other words, the system is able to learn. This approach is called *Explanation-Based Learning* (*EBL*). An evolution of the *EBL* is proposed in [12]. This approach tries to merge the old *EBL* engine, based on symbolic knowledge representation, with the statistical approach. The proposed system aims to take advantage of the robustness of statistical approach respect to real word problems, but at the same time it exploits the expressive power of symbolic knowledge representation.

An alternative approach to generalisation uses formal languages, and is known as *syntactic pattern recognition* [13]. In these systems, concepts are decomposed into simpler parts and their description relies on a grammar. A grammar is formally defined by the quadruple (Σ, N, P, S), where:

- Σ, the *alphabet*, is the set of the so-called terminal symbols, i.e. the basic elements of the grammar;
- N is the set (disjoint from Σ) of the nonterminal symbols; each of these symbols represents one or more strings of terminal and nonterminals symbols.
- P is the set of the *production rules*, composed by a head, represented by a nonterminal, and a body made up of a sequence of terminals and/or non terminals.
- $S \in N$ is a special symbol, known as the start symbol.

The set P represents possible and interesting structures, i.e. frequent patterns. The problem of inferring knowledge is stated as the problem of design a learning machine for pattern recognition, where a pattern is a particular structure included into the grammar. The system infers a grammar from training examples and applies it on the new data, in order to verify if the string of terminal symbols belongs to the learned grammar. This kind of approach requires preliminary work by the designer in ontology domain definition, in order to identify the key elements of the representation. The major drawback with this methods is the high computational cost needed to infer grammars. Historically, these approaches has been considered as alternatives to statistical learning systems, but during last decades many efforts have been made to unify statistical and syntactic pattern recognition (see [14]).

Other authors consider traditional approaches inadequate to cope with the complexity of managing knowledge and its evolution in complex phenomena. However, they believe that these scenarios cannot be modeled only by mathematical or statistical means. For example, Evolving Transformation System (*ETS*) is a formalism that tries to unify the syntactic and statistical pattern recognition, in order to create a new kind of class representation. The definition of *class*, according to the author, rests on the generative side: objects belonging to the same class share similar generative histories. In this context, a generative system is a nondeterministic system operating on actual entities and assembling them into larger entities (and eventually into class objects), guided by some hierarchical description of the class [15]. This kind of representation is focused on the problem of giving a structural representation to the data. Each object in this formalism is thought of as a temporal structural process and the representation of each element of a class evolve with the description of the class itself. *ETS* is a work in progress framework, limited by the lack of new mathematical instruments to deal with the complexity of a structural description.

In [16], Chazelle proposes a new vision to deal with phenomena arising from life sciences, stating that means used in physical science are not adequate. According to his work, algorithms are more suitable for these purpose, due to their rich and expressive language. Moreover, the author claims that some problems can take an enormous advantage from the novelties introduce by a new perspective, taking into account the peculiarities of complex non physical systems. In the case of sensory data used to investigate and predict human habits and behaviour, the complexity is very high, because of the high number of variables to include in the model. Chazelle introduces the *natural algorithms* to model these systems. This approach relies on the so-called *influence systems*, i.e. networks of agents that perpetually rewire themselves. These networks are specified by two functions: f and G; the function f calculates the position of an agent, taking as input the location of its neighbour agent, given by function G. The output of G is function of the state of the whole system, that is the position of all agents. In this approach is possible to note how the information travels through the system, in a way that separates its syntactic or structural component and its semantic. In other words,this method models complex systems exploiting equally qualitative and structural information.

Our proposal differs from those presented in literature, because it aims to extract knowledge from a large set of unstructured data (such as sensory data), translating it in a machine-understandable form. Many approaches have attempted to deal with the complexity of such kind of data. In particular, many systems have been proposed in the area of Ambient Intelligence, which typically deals with sensor readings, and their interpretation. For example, in [17], the authors suggest a three-tier paradigm for knowledge extraction. In particular, this paradigm cuts irrelevant details off from raw sensor readings, in order to obtain more refined data that can be analyzed by the reasoning module, at the top of this processing hierarchy. Statistical methodologies (e.g., correlation analysis, clustering) are used in [18] to cope with the complexity of large sensor reading dataset. The proposed system models and learns user habits through his interactions with the actuators deployed in the environment. According to the authors, user habits are coded into sensor readings, thus they can be inferred analyzing sensory data and discovering relations between environmental conditions and user.

Our system uses a similar approach, but, at the same time, aims to use as little a-priori knowledge as possible, because this is hard to obtain in the great part of real problem belonging to sensory data mining. In fact, such data is hardly understandable according to simple user description of the phenomenon that has generated it. So, it is very hard to formulate a-priori hypotheses, as in deductive approaches; this may force to use a too specific and detailed model, with a high risk of overfitting. On the other hand, an expert of the application domain possesses some knowledge, which can represent a very important resource. The actual problem is translating it to be usable by the system. With a pure inductive approach, this task would include a very hard empirical work, in order to tune up all the parameters used in the chosen method. In this case, the risk of overfitting is very high, because the relations between sensory data and model parameters are not very clear, so it is impossible to distinguish when the over fitting starts. A structural approach is less prone to this problem, because it can be more simple build measure of complexity over the model it uses, due to its representation.

3 Ontology Learning

The most common ontology definition can be found in [4]; the author defines an ontology as the specification of a conceptualization, that is a model of the concepts and relations describing a particular domain. According to [6], this definition can be formally translated in a structure $\mathscr{O}$:

$$\mathscr{O} := (C, \leq_C, R, \sigma_R, \mathscr{A}, \sigma_A, \mathscr{T})$$

- four disjoint sets: concept identifiers (C), relation identifiers (R), attribute identifiers ($\mathscr{A}$), data types ($\mathscr{T}$);
- a semi-upper lattice $\leq_C$ on C;
- a *relation signature* function $\sigma_R : R \rightarrow C^+$;
- a *relation hierarchy* partial order $\leq_R$ on R;
- a function $\sigma_{\mathscr{A}} : \mathscr{A} \rightarrow C \times \mathscr{T}$.

Thus, learning an ontology means specifying all these elements, by inferring them from data. This problem can be decomposed into the following subtasks:

1. acquisition of the relevant terminology;
2. identification of synonym terms;
3. formation of concepts;
4. hierarchical organization of the concepts (concept hierarchy);
5. learning relations, properties or attributes, together with the appropriate domain and range;
6. hierarchical organization of the relations (relation hierarchy);
7. instantiation of axiom schemata;
8. definition of arbitrary axioms.

This schema has been formulated mainly in the context of ontology learning in text corpora. Obviously, in the context of sensory data these steps need some changes, in order to adapt them to the different characteristics of the new scenario. However, the general structure of the process remains the same. Thus, in this work we propose an adaptation of this schema, modifying it according to the new kind of data and the differences between text and sensory data.

Undoubtedly, the goal we aim at is very challenging, and many issues are to be addressed; some of those are related to theoretical open issues in computer science, so it is impossible to known if they are practically solvable. We do believe, though, that knowledge extraction can take advantage of an ontology learning approach, because this might exploit the information potential embedded in sensory data. The basic idea is that data collected from sensors share an underlying language, i.e. it can be considered as generated by a particular language describing some phenomena. Similarly to what described in [19], we assume that data are drawn from a process that can be modeled by a Turing Machine (TM). This means, according to Chomsky, that there is a language that can describe such data. Likely, this language is very complex and, moreover, data is corrupted by noise, so reconstructing the original language from data is a very tricky task.

If we compare the problem of grammar induction (i.e., learning) and the problem of ontology learning, we discover that they have much in common. This can be explained if we consider the ontology problem as the semantic side of grammatical induction. The literature has shown that this two aspects are strictly coupled and it is impossible to solve one of them discarding the other.

Given these caveats, it is simple to describe and motivate the changes we made to the general ontology learning schema. From a sensory data point of view, we are not very interested in the relation discovery and our focuses is principally on concepts and their hierarchy.

The proposed approach is composed by:

1. individuating a set of basic properties (axioms) and features to discovery significant patterns;
2. discovery of relevant elementary patterns as terminology;
3. abstraction of patterns as concepts;
4. inferring hierarchical concept organisation;

Setting axioms In the first step, the key elements of the ontology are defined; it is the only phase of the system that requires human intervention. The analyst has to specify a description of the goal concepts in terms of general properties, like time, space or other very general features. The main difference from others approaches is how these features are described. For example, consider the user activity recognition task. A user activity may be defined as *a recurrent sequence of actions, that can be recursive decomposed in simpler subtasks*. Two approaches can translate this definition in features on data: the deductive, and the inductive one. According to the deductive approach, the definition is transformed into an abstract and general model, that hypothesises sensor reading interactions that identify executions of the same activity. In the inductive approach, the analyst translates the definition in terms of properties the sensors can measure, such as time duration; so, an activity is treated as a recurrent pattern in data, whose instances have similar structures and time durations. No attempts at generating a general activity model are made, but the model will emerge from data. The feature selection depends on the experience of the analyst, but it is a simpler task than the other approaches. Moreover, it is less prone to error and axioms chosen can be simpler checked.

Discovery terminology The second step faces the problem of finding data with the properties defined at the previous step. There exist several techniques to accomplish this task, but a data mining approach is the best choice. In fact, a statistical analysis of data can aid the most significant pattern to emerge, so the basic level of the ontology can be identified with the smallest, but more recurrent patterns in sensory data. Moreover, these techniques guarantee robustness, reliability, and thoroughness, since they have been extensively used in many different systems in the last decades. The a-priori based algorithms are an example of these techniques.

This step can be thought as a data fusion, i.e. the system associates data coming from different source and of different types. This is a very common approach in this kind of systems (e.g., [20, 21]) and it is useful to process data in a multi-sensor context, in order to explode relations between different sensor triggers.

Pattern abstraction Pattern abstraction allows to obtain a generalization from the instances of patterns present in data. The system addresses the problem of *synonyms*, grouping similar instances of the same patterns. In particular, the best choice for this step is the use of a technique coming from the statistical learning. In fact, these approaches allow a statistical description of data, summarising it according to the most emergent properties. Unsupervised clustering algorithm and subsequent statistical classifier can be used in order to discover and then recognise most common patterns. This kind of systems are well-known in literature and have obtained a successful application in many research area. So, at the end of this step the system will be able to associate a statistical model to each pattern and a trained classifier to distinguish between them.

Inferring concept hierarchy In this step, the system finds recurrent terminology structures. A grammar induction algorithm is used: the system considers the data as a language and looks for the suitable grammar to generate it. According to [22], a grammar can be defined as a device that enumerates the sentences of a language. In other words, it is a function, F, that generates every sentence of a language, $\mathcal{L}$. The expressive power of these devices depends on the restrictions imposed on F. If no restriction is imposed, then the function belongs to the set of General Turing Machine. On the contrary, if a strict restriction on F is imposed, such as the constraint on each grammar to be a finite Markovian source, then it is possible to prove that some language (e.g., natural language) can not be generated by these grammars. So, it is important to choose the adequate restriction, based on the aim set for the language. In this context, we assume that a grammar represented by a Finite Automaton may have the sufficient power to model sensor language.

Representing each base concept, that is each term, with a terminal symbol of a grammar, it is possible to reconstruct complex concepts as sentences of the same language. In fact, once the system infers a grammar from data, the syntactic representation of a complex concept is represented by the derivation tree of the sentence.

In [23], the task of grammatical inference is widely debated. Many learning system have been proposed and each of them is specialized for a particular target, with respect to the kind of data and the application domain. Despite most systems focus on learning grammar from text, there are others that have been applied to bioinformatics or computer vision. This means that grammatical inference is possible even if data are less structured than text data and results obtained in this area encourage the use of these techniques in the scenario of sensory data.
Moreover, a representation of a sensory data through a grammar can be seen as a compression of the data. The Occam's Razor gives us a simple guideline to choose between the possibly different grammars that can represent the data equally well. In fact, the simplest explanation is always the best, so the shortest grammar is the best to represent data. Measures of complexity in the sense of Occam's Razor are Minimum Description Length (MDL) principle [24] or Kolmogorov complexity [25]. Even if it is impossible to calculate the MDL or similar measures, they represents a good guideline for the choice among different representations.

4 A Proof of Concept: The Slide Puzzle

Extracting knowledge from sensory data is a very complex and difficult challenge and may prove not too useful to test the effectiveness our approach for the present discussion. In fact, a too complex problem does not allow to study the details of every single part of the system and their effects on whole results. A more manageable problem, whose difficulties can be tuned according to very specific tasks, appears a more reasonable choice. Thus, a very similar problem was chosen to test the proposed system, even though it is still representative of the original target scenario, due to their very similar features. Extracting knowledge from sensory data shares in fact many similarities with the heuristic learning problem, because the success of both of them relies on the ability to discover hidden and counter-intuitive relations in data.

In fact, heuristic learning is a very interesting research area in structural analysis. In some cases, it is difficult to unveil the structure of a problem, in order to improve solution search algorithms. This is the case, for instance, of the n-puzzle slide game.

The n-puzzle problem is a generalisation of the more common 15-puzzle. In its original form, the puzzle consists of 16 squares numbered from 1 to 15 and arranged in a 4×4 box, with one position of the box left empty. A legal move consists of sliding an adjacent block into the empty space. The goal is to reposition the squares from a given arbitrary starting arrangement by sliding them one at a time into the configuration shown in Fig. 1.

Ratner and Warmuth [26] showed that finding an optimal solution to n-puzzle, i.e one involving as few moves as possible, is an NP-complete problem. Moreover, getting the goal position, that is solving the puzzle, is not possible for all initial arrangements. Only instances corresponding to even permutations of the goal configuration are solvable, that is only half of all possible configuration can be transformed into the goal configuration (setting as initial configuration the one with the empty tile on the

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Fig. 1 Goal position of 15-puzzle

bottom right corner). The 8 instance of the problem has been solved with a breadth-first search, and the 15 one has been solved with IDA* search algorithm [27]. The heuristic used by this search algorithms are based on Manhattan distance, but there is no suggestion as regards a general solution algorithm, i.e. one that is independent of the dimensionality of the problem. For higher dimension instance of the problem, such as 24-puzzle, many ad-hoc solutions were used, based on the particular properties of the problem. The main difficulty with high dimensional instances of n-puzzle concerns the very wide search space, that make an efficient search impossible.

Cutting-edge research has been devoted to finding a suitable solution to the 24- and higher instance of this game. The first proposed solution used some particular features of the problem to speed-up search and obtaining an optimal solution [27]. In particular, this approach encode more knowledge of the problem in the form of improved heuristics, in an automatic fashion. The drawback with this system is that its heuristic does not guarantee a good execution time on every instance of the problem, but some random instances take very long time to be solved or worse they have not been solved at all. Later tries to solve the problem have been turned on the problem of finding better heuristics, created with the aid of machine learning. Discarding the strong limitation of finding admissible heuristics, Ernandes and Gori proposed a machine learning approach using an Artificial Neural Network. Given a rich set of training example, *ANN* learns how far a configuration, i.e. a node, is from the goal state. This approximation is used to improve the heuristic, with a remarkable speed up for the search in the 8 and 15 instances of the game. In the case of 24-puzzle, the lack of a good training set make an effective use of the *ANN* impossible. The authors proposed but did not implement a bootstrapping technique to overcome this problem. That is, they assert that is possible to specialise an initially weak heuristic function in an iterative manner, using example generated at each step by each function. The authors of [28] continued the work by Ernandes and Gori, implementing the bootstrapping approach. In particular, they used incremental bootstrapping process augmented by a random walk method for generating successively more difficult problem instances, obtaining good results in the solution of random generated instances of 24-puzzle. The great drawback with this approaches is the long time the bootstrapping phase takes. The author proposed a solution only in the case of a single instance problem: in that case the bootstrapping are interleaved with a classic heuristic search to lower total execution time. A similar approach can be found in [29]. The authors propose a system that learns a heuristic to solve a single instance, and the bootstrapping is done over successive failed attempts to solve the instance. They show the effectiveness of their approach in the 15-puzzle game.

In [30] a system is proposed that mixed multiple heuristics to improve them. In this work, the key idea is to merge the knowledge coming from different heuristic, merging them into a better one that summarise all the others. This knowledge mesh-up is produced by an *ANN*, that is by a statistical learning. A very interesting work is that presented by Korf and Felner in [31]. The authors deal with the problem of finding an accurate admissible heuristic. They propose a memory-based approach, that utilises partial puzzle resolution to obtain an estimate of the cost. The scheme is dived into many parts, and for each a number of move to goal state is saved in a database. With a carefully choose of the subparts, is possible to obtain a good estimate and therefore a tight (and better) heuristic. The authors obtain relevance results on the 24-puzzle, proving the goodness of their idea. Also, this means that the recurring relations and interactions between subproblems are the key of this problem, and thus capturing them leads to a better solution.

4.1 Hidden Relations: Knowledge to Enhance Heuristic

The work produced in this area shows as the solution of n-puzzle game, with $n > 15$ requires a heuristic enriched with some knowledge inferred by example of simpler instance solutions. In our opinion, this means that the proposed systems are trying to summarise some unveiled properties and relations of the application domain to empowered search algorithm, as demonstrated by the use of machine learning approach and in particular of statistical learning (e.g., *ANN*).

We agree with this vision and we proceed a step further: the system has to learn the structure of known solutions and has to find relevant relations between them in order to go beyond combinatorial explosion nature of the problem. Obviously, we do not assert that we are able to find an optimal solution for the problem, but we only want to drive our system to "comprehend" the essence of the game and could suggest a possible heuristic for a higher dimensionality game, focusing computational efforts especially on more promising areas of the search space. This implies the identification of recurrent structure that constitute the sub problem to solve and a model to describe how they interact.

The basic idea is to describe a solution as the evolution of a set of properties evaluated on each state it goes through. Each of these properties expresses a particular feature related to the state. It can be a static feature, such as the Manhattan distance to the goal solution, or a dynamic one, i.e. the row of the last moved tile. We suppose that a good set of properties can capture some general structures shared by solutions. This choice is similar to the multiple heuristic approach presented in [30], and is motivated by the same reasons: we are trying to mix different properties in order to unveil not evident and counter-intuitive features of the solution.

Given a set of solutions to the 8-puzzle, 15-puzzle and some simple instances of the 24-puzzle, we can translate them in their feature evolution representation.

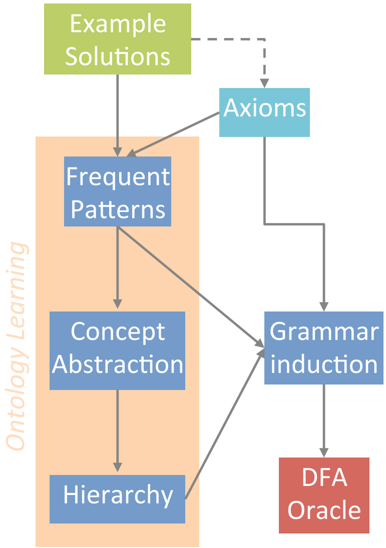

Fig. 2 Sketch of the proposed system

Assigning a unique symbol to each possible feature combination, we can describe a solution as a string made up of these symbols. In this way, we can describe each solution as a string generated by the language of solutions. In this language, we the terminal symbol are represented by the unique symbol chosen for each state.

In the next step, we try to obtain the *Discrete Finite Automaton* (DFA) that recognize that language. In this manner, we have found a machine able to suggest us the most probable path in the tree to find the solution. At each node, we can evaluate the solution we are building and choose next step as that most probable according to our DFA. The DFA acts like an oracle, able to guide in the path toward the solution.

In Fig. 2, the proposed system is shown. The two main blocks of the system are the *Ontology Learning* and the *Grammar Induction*. According to the previous description of the Ontology Learning approach, proposed in Sect. 3, the main steps of this process are presented in the figure. The first step, that is stating axioms, is the only non fully-automated step of the process and, therefore, it is placed outside of the *Ontology Learning*: domain expert's analysis of example data is required in order to obtain axioms. The whole process of *Ontology Learning* produces the input data for the Grammar induction module. In particular, terminal symbols (axioms), an initial approximations of nonterminals (concepts) and production rules (hierarchy), are elaborated by the *Grammar induction* module to produce the *DFA Oracle*.

At the end of the processing chain, the system builds up the oracle, and an insight into the structure of input data that has generated it. This ontology can be used to evaluated the quality of the solution and to suggests to the designer new tuning to the axioms definition, in order to capture the actual structure of the problem.

5 Conclusion

In this chapter we presented a proposal for a system to cope with the complexity of knowledge managing, sharing and reuse in the sensory data scenario. Nowadays, this is a very hard task, due to the great amount of raw data and to their lack of evident structure.

Unlike other systems presented in literature, we try to infer an ontology directly from data and using it to discover new intelligible knowledge. The system we propose uses a structural approach, based on the use of grammatical inference, in order to emerge the most relevant relations present in data.

We propose the heuristic search as a controlled testbed for our system. In particular, we measure the potential of sour approach in the n-puzzle game.

Acknowledgments This work is partially supported by the PO FESR 2007/2013 grant G73F11000130004 funding the SmartBuildings project and by the PON R&C grant MI01_00091 funding the SeNSori project

References

1. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
2. Cook, D., Youngblood, M., Heierman E.O., Gopalratnam, K., Rao, S., Litvin, A., Khawaja, F.: Mavhome: an agent-based smart home. In: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), pp. 521–524, (2003)
3. Doctor, F., Hagras, H., Callaghan, V.: A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. IEEE Trans. Syst. Man Cybern. (Part A: Systems and Humans) **35**(1), 55–65 (2005)
4. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2), 199–220 (1993)
5. Velardi, P., Faralli, S., Navigli, R.: Ontolearn reloaded: A graph-based algorithm for taxonomy induction. Comput. Linguist. **39**(3), 665–707 (2012)
6. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, New York (2006)
7. Yang, Q., Wu, X.: 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making (IJITDM) **5**(04), 597–604 (2006)
8. Tukey, J.W.: Exploratory Data Analysis. Addison-Wesley, Menlo Park (1977)
9. Hájek, P., Havránek, T.: Mechanizing hypothesis formation : Mathematical Foundations for a General Theory (Universitext). Springer, New York (1978)
10. Mitchell, T., Keller, R., Kedar-Cabelli, S.: Explanation-based generalization: A unifying view. Mach. Learn. **1**(1), 47–80 (1986)
11. Dejong, G., Mooney, R.: Explanation-based learning: an alternative view. Mach. Learn. **1**(2), 145–176 (1986)
12. DeJong, G.: Toward robust real-world inference: a new perspective on explanation-based learning. In: Proceedings of the 17th European conference on Machine Learning (ECML'06), pp. 102–113. Springer, Heidelberg (2006)
13. Fu, K.S.: Syntactic Methods in Pattern Recognition, Mathematics in science and engineering, vol. 112, Academic press, New York (1974)

14. Tsai, W.H., Fu, K.S.: Attributed grammar-a tool for combining syntactic and statistical approaches to pattern recognition. IEEE Trans. Syst. Man Cybern. B Cybern. **10**(12), 873–885 (1980)
15. Goldfarb, L.: Representation before computation. Nat. Comput. **9**(2), 365–379 (2010)
16. Chazelle, B.: Natural algorithms and influence systems. Commun. ACM **55**(12), 101–110 (2012)
17. De Paola, A., Gaglio, S., Lo Re, G., Ortolani, M.: An ambient intelligence architecture for extracting knowledge from distributed sensors. In: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, ACM, pp. 104–109, (2009)
18. Augello, A., Ortolani, M., Lo Re, G., Gaglio, S.: Sensor mining for user behavior profiling in intelligent environments. In: Advances in Distributed Agent-Based Retrieval Tools, pp. 143–158. Springer, Heidelberg (2011)
19. Gaglio, S., Lo Re, G., Ortolani, M.: Cognitive meta-learning of syntactically inferred concepts. In: Samsonovich, A.V., Johannsdottir, K.R. (eds.) BICA, Frontiers in Artificial Intelligence and Applications, vol. 233, pp. 118–123. IOS Press, Amsterdam (2011)
20. De Paola, A., Cascia, M., Lo Re, G., Morana, M., Ortolani, M.: User detection through multi-sensor fusion in an ami scenario. In: 15th International Conference on Information Fusion (FUSION 2012), pp. 2502–2509, (2012)
21. De Paola, A., Gaglio, S., Lo Re, G., Ortolani, M.: Multi-sensor fusion through adaptive Bayesian networks. In: AI* IA 2011: Artificial Intelligence Around Man and Beyond, pp. 360–371. Springer, Berlin (2011)
22. Chomsky, N.: On certain formal properties of grammars. Inf. Control **2**(2), 137–167 (1959)
23. de la Higuera, C.: Grammatical Inference: Learning Automata and Grammars. Cambridge University Press, New York (2010)
24. Rissanen, J.: Minimum description length principle. In: Encyclopedia of Machine Learning. Springer, New York (2010)
25. Li, M., Vitányi, P.M.: An Introduction to Kolmogorov Complexity and Its Applications, 3rd edn. Springer, Heidelberg (2008)
26. Ratner, D., Warmuth, M.K.: Finding a shortest solution for the n $\times$ n extension of the 15-puzzle is intractable. In: Proceedings of the 5th National Conference on Artificial Intelligence (AAAI), pp. 168–172, (1986)
27. Korf, R.E., Taylor, L.A.: Finding optimal solutions to the twenty-four puzzle. In: Proceedings of the 13th National Conference on Artificial intelligence (AAAI'96), Vol. 2, pp. 1202–1207. AAAI Press, Menlo Park (1996)
28. Arfaee, S.J., Zilles, S., Holte, R.C.: Learning heuristic functions for large state spaces. Artif. Intell. **175**(16–17), 2075–2098 (2011)
29. Humphrey, T., Bramanti-Gregor, A., Davis, H.: Learning while -solving problems in single agent search: Preliminary results. In: Gori, M., Soda, G. (eds.) Topics in Artificial Intelligence, Lecture Notes in Computer Science, vol. 992, pp. 56–66. Springer, Heidelberg (1995)
30. Samadi, M., Felner, A., Schaeffer, J.: Learning from multiple heuristics. In: Proceedings of the 23rd National Conference on Artificial intelligence (AAAI'08), vol. 1, pp. 357–362. AAAI Press, Menlo Park (2008)
31. Korf, R.E., Felner, A.: Disjoint pattern database heuristics. Artif. Intell. **134**(1–2), 9–22 (2002)