Rule Extraction from Support Vector Machines: An Introduction

Joachim Diederich

American University of Sharjah, UAE and University of Queensland, Australia

Rule extraction from support vector machines (SVMs) follows in the footsteps of the earlier effort to obtain human-comprehensible rules from artificial neural networks (ANNs) in order to explain "how" a decision was made or "why" a certain result was achieved. Hence, much of the motivation for the field of rule extraction from support vector machines carries over from the now established area of rule extraction from neural networks. This introduction aims at outlining the significance of extracting rules from SVMs and it will investigate in detail what it means to *explain the decision-making process of a machine learning system* to a *human user* who may *not be an expert* on artificial intelligence or the particular application domain. It is natural to refer to both psychology and philosophy in this context because "explanation" refers to the human mind and its effort to understand the world; the traditional area of philosophical endeavours. Hence, the foundations of current efforts to simulate human explanatory reasoning are discussed as are current limitations and opportunities for rule extraction from support vector machines.

1 Explanation: The Foundations

In a series of paper and books, Paul Thagard explores what it means to explain something (most recently Thagard and Litt forthcoming). Human thinking is essentially an ongoing, inner dialogue to explain why certain events do or do not happen or why things behave in a certain way. Explanation is closely linked to problem solving because the failure to explain an event or a certain outcome may trigger a problem solving episode. People explain to themselves and others why things are not working properly and what to expect if certain actions are taken. Explanation is a continuous cognitive process almost identical to thinking because humans are constantly explaining "why" things happen and "how" things work.

J. Diederich: *Rule Extraction from Support Vector Machines: An Introduction*, Studies in Computational Intelligence (SCI) **80**, 3–31 (2008)
www.springerlink.com

1.1 Forms of Explanation

It is useful to distinguish between various types of explanation, e.g. causal explanations that are acceptable answers to "why" questions as opposed to the step-wise explanations that are acceptable responses to "how" questions. It is also possible to ask for clarifications if certain facts are known already and more detail is required. Finally, there are those types of questions that are best answered by providing an example for an event or fact that is most typical and therefore helps to explain a whole set of observations.

Thagard and Litt (forthcoming) distinguish between three major explanatory processes:

- Providing an explanation from available information
- Generating new hypotheses that provide explanations
- Evaluating competing explanations

The four major theoretical approaches are: "*deductive*, using logic or rule-based systems; *schematic*, using explanation patterns or analogies; *probabilistic*, using Bayesian networks; and *neural*, using networks of artificial neurons" (Thagard and Litt forthcoming, p. 2).

The classical explanation is deductive and requires logical reasoning. What is to be explained (the explanatory target) follows from known facts by logically applying a set of rules ("Anyone with influenza has fever, aches and cough. You have influenza. So, you have fever, aches and cough", Thagard and Litt forthcoming, p. 4). This is the modus ponens, a simple, logical argument: If X is true then Y is true. X is true. Therefore Y is true.

Sometimes the explanatory target is only probable and more than one explanation is possible. Explanation is then closer to a conditional probability. Often, it is useful or required to find the best explanation which can be a complex process because whole "explanation structures" need to be examined to determine the best (Thagard, 1978).

Logical, deductive explanation requires a set of known facts as well as a set of IF ... THEN rules. Background knowledge is then expressed in the form of propositions which are used to explain an explanatory target. Here is a simple example following (Thagard and Litt forthcoming):

1. Anyone who completes a marathon has muscle pain and feels tired.
2. Person X has muscle pain and is tired.
3. Person X has completed a marathon.

The conclusion here obviously does not necessarily hold: there are many reasons why a person may have muscle pain and may feel tired, running a marathon is just one of them and maybe not the most probable explanation. Hence it is possible to have a loser association between propositions and an explanatory target. Again following the general argument in Thagard and Litt (forthcoming), it is possible to characterize *causal schemas* as an alternative to formal deduction. Here is a simple example:

1. Explanatory pattern: Typically, running a marathon causes muscle pain and tiredness.
2. Explanatory target: Person X has muscle pain and is tired.
3. Schema instantiation: Maybe person X has completed a marathon.

The example above invites already the application of probability theory and statistics. Again in the context of our simple example: the probability of muscle pain and the feeling of tiredness after a marathon is high. The marathon explains why person X feels pain and is tired. Here explanation is more like a conditional probability and the value of the explanation depends on known probabilities and the match with the schema.

Thagard and Litt (forthcoming) outline an additional way of modelling explanation and interestingly this is the use of artificial neural networks. Thagard and Litt (forthcoming) confirm that the neural approach by itself is not a theory of explanation; it is a method that simulates the cognitive processes that are part of explanatory thinking. Thagard and coworkers have used neural networks for cognitive modelling including the generation of explanations. In the current context, it has been demonstrated again and again that in particular feedforward neural networks are lacking explanatory power and hence rule extraction has to be applied. This is our point of departure from Thagard's argument.

In summary, the classic review is that explanation is a deductive argument including background knowledge and inference rules such as modus ponens. The inference rules allow the sequential application of "if-then-else" statements in order to justify an explanatory target. Whenever no precise knowledge is available, explanatory schemas or probabilistic rules can be used. But of course other forms of explanation are possible as well.

1.2 Analogy as a Form of Explanation

Every classroom teacher knows that at times it is very difficult to introduce a new theoretical concept. Even with the best of efforts it may not be possible for the class to grasp the theoretical elements that are being introduced. A single example, however, may change all of that and leads to an "aha" experience and complete understanding of the new material. So examples do have explanatory value and can be most useful, in particular if they are typical or even prototypical. In the current context, that his rule extraction from support vector machines, it may well be an objective to identify one or more examples that explain the behaviour of the machine (see Martens et al. and Nunez et al. in this volume). If this is not possible and precise background knowledge in the form of rules is not available, an *analogy* may be used for explanation.

An analogy requires the existence of a memory system to store and search cases which may have varying degrees of similarity. For instance, a case-based system could store various types of sporting events, including those that

require a great deal of endurance. There are obviously similarities between a marathon and a triathlon and it is to nobody's surprise that both can cause tiredness and muscle pain immediately after the event. Even if we don't know that a particular person has just completed a marathon, the similarities between a long run and a multi-sports event explain why a marathon runner should be just as tired as a triathlete. Here, explanation is based on obvious and implicit similarities.

Analogies and schemas go hand in hand. Instead of storing all sort of sporting events we can have one schema for endurance sports which includes the immediate consequences including muscle pain and tiredness. For any given sports of this type, special features are replaced by variables which can be instantiated whenever an explanation is required. There could be a variable "physical effects" which can be instantiated by the two known consequences of endurance sports.

In summary, the following processes can generate explanations acceptable to humans:

- Logical deduction by use of inference rules
- Probabilistic rules including conditional probabilities
- Schemas based on the similarity between cases
- The provision of one or more examples which are typical or even prototypical

In artificial intelligence, several of these processes are often combined to arrive at systems that either use or generate explanations. The best example is *explanation-based learning* or *explanation-based generalization.* This form of learning has a long history in cognitive science and is often traced back to *Gestalt* theory, a branch of psychology popular in continental Europe in the first half of the twentieth century. Gestalt theory in turn has its roots in a rational form of philosophy which assumes that a significant part of our knowledge is innate and learning occurs at the periphery of knowledge only. That is, a significant amount of background knowledge is required for successful learning. On the other side, very few examples are necessary.

1.3 Explanation-Based Generalization

Explanation-based generalization is most interesting for the discussion here because it uses logical deduction based on the presentation of a single example. Since its invention twenty years ago, it has been modified to allow probabilistic reasoning and even the use of schemas. Explanation-based generalization had a significant impact on theory formation in artificial intelligence and early efforts to realize rule extraction from neural networks are linked to the attempt to build explanation-based learning systems. Hence, explanation-based generalization had a significant impact on the current understanding of what constitutes an explanation and will be briefly summarized here.

An explanation-based generalization system requires four components (1) the target concept, (2) the training example, (3) background knowledge in the form of rules and (4) the operationality criterion which defines what is to be learned or how learning should improve performance. There is an additional requirement that the training example is a positive instance of the target concept. Also, background knowledge must be both complete and consistent. Explanation-based generalization proceeds in two stages (1) explanation and (2) generalization. During the first stage a formal proof is constructed which demonstrates that the training example is a positive instance of the goal concept. If this is not the case, learning can fail. In the second stage, on the basis of the formal proof that the example is a positive instance of the goal concept, a new rule is formed and the knowledge is added to the rule base. Most importantly, and in contrast to statistical learning systems such as neural networks and support vector machines, the generalization that is the result of the learning episode is justified: it can be formally proven that the generalization holds given the training example, the goal concept and the background knowledge.

An example may be useful to explain explanation-based generalization: assume you are walking the streets and you see a car you have not seen before but you recognize as a BMW. The car has features you associate with BMWs but there are new, unexpected aspects as well. These new features immediately generate interest and start a learning process that is indeed an explanation process: the known attributes (this may be "shape of head lights", "company logo", etc.) are being used to explain that this particular car is a BMW. This is the first stage of explanation-based generalization. The process may continue with a generalization: A new class or concept is added to the background knowledge that includes the just seen car but maybe used to recognize other cars identical or similar to this one. As a matter of fact, this scenario invites a generalization as it is extremely unlikely that the just seen car is unique and no others of this type exit.

Explanation-based generalization is a very natural model for human learning. The previous paragraph describes an everyday observation and the learning process that is triggered by an observation. In addition, explanation-based generalization uses a "proof structure" (the sequence of rules that have been applied to prove that the training example is an instance of the goal concept) as the basis of an explanation. Hence, in the following chapters explanations are "rule sets" that explain "how" a certain decision was made and "why" it was made.

1.4 How and Why Explanations

As indicated earlier, it is useful to distinguish between various types of explanation, e.g. causal explanations that are acceptable answers to "why" questions in contrast to the step-wise explanations that are acceptable responses to "how" questions. Let's focus on how questions first.

A how explanation consists of a sequence of rules that map a given input to an output, in this case the input to a neural network or a support vector machine. In explanation-based generalization, a sequence of rules explains how the single training example is a positive instance of the target concept. Even though both rule extraction from neural networks and support vector machines generate rule sets, these rules are rarely applied in sequence. Very often rules are propositional in nature and include sets of inputs that result in a positive output. In this sense, rules can be independent and can even overlap. In rule extraction from neural networks, each rule includes a set of inputs that can result in a positive or negative output independently. The totality of the rule set explains how the neural network arrives at a decision.

"Why" explanations are typically used in expert systems. Here, the user may be engaged in an extensive dialogue and sometimes the system poses questions which are difficult to understand. After each question, the user has the option to ask "why" and the system will justify asking that particular question at this point in time. In this sense, "Why" explanation in expert systems does not include deep causal reasoning or the identification of a limited set of inputs that causes some output. A justification for a question is given.

1.5 Generating or Identifying the Best Explanation

Very often, more than one explanation is possible and explanations may even compete. Thagard and Litt (forthcoming) identify the evaluation of competing explanations as one of three major processes modelled by computational systems that aim to simulate human reasoning. The term "abduction" is well established in artificial intelligence and describes the inference to the best explanation as well as the generation of hypotheses (Thagard and Litt forthcoming). Thagard and Litt (forthcoming, p. 9) identify three criteria for the best explanation:

- Consilience: How much does a hypothesis explain?
- Simplicity: How many additional assumptions are required to carry out an explanation?
- Analogy: Are there hypotheses whose explanations are analogous to accepted ones?

2 Rule Extraction from Support Vector Machines: Aims and Significance

Andrews et al. (1995) describe the motivation behind rule extraction from neural networks. The five points outlined below, with the possible exception of "knowledge acquisition for expert systems", are relevant for the current effort to extraction comprehensible rules from SVMs. A brief review of Andrews et al. (1995) arguments will help to establish aims and significance for rule

extraction from SVM techniques. For an introduction to support vector machines, see Martens et al. in this volume.

2.1 Provision of a "User Explanation" Capability

In symbolic artificial intelligence (AI), the term *"explanation"* refers to an *explicit structure* which can be used internally for reasoning and learning, and externally for the explanation of results to a user. Users of symbolic AI systems benefit from an explicit declarative representation of knowledge and traditionally, symbolic AI systems are deductive techniques: Reasoning (including classification) is from the *"generic"* (expressed in the form of general rules) to the *"specific"* (an instance or individual that is to be classified). Even learning, if it is based on a large amount of background knowledge, is deductive in symbolic AI systems.

The explanation capability of symbolic AI is based on intermediate steps of the reasoning process, e.g. a trace of rule firings, a proof structure, etc., which can be used to answer *"How"* questions. Gallant (1988) observes that the benefits of an explanation capability include a check on the internal logic of the system as well as enabling a novice user to gain insights into the problem at hand.

An explanation capability is considered to be one of the most important functions provided by symbolic AI systems. The ability to generate even limited explanations is essential for the user-acceptance of such systems (Davis et al., 1977). In contrast to symbolic AI systems, neural networks have no explicit declarative knowledge representation, and with exception of structured connectionist systems, neural networks do not perform deduction. Therefore neural networks have considerable difficulty in generating explanation structures and the situation is no different in support vector machines.

Traditionally, practitioners in the field of symbolic AI have experimented with various forms of user explanation, in particular *rule traces* (i.e. the sequence of rules or inference steps that are part of a problem-solving episode). However, it is obvious that explanations based on rule traces are too rigid and inflexible (Gilbert, 1989) because rules may not be equally meaningful to the user. In addition, rule traces always reflect the current structure of a knowledge base. Further, rule traces may have references to internal procedures (e.g. calculations); might include repetitions (e.g. if an inference was made more than once); and the granularity of the explanation is often inappropriate (Gilbert 1989; Andrews et al. 1995). A clear lesson from the use of rule traces in symbolic AI is that the transparency of an explanation is by no means guaranteed. For example, an explanation based on rule traces from a poorly organised rule base with perhaps hundreds of premises per rule cannot be regarded as *"transparent"*. Interestingly, it is an inherent problem of rule extraction from neural network techniques (in particular those that are learning-based), that a large number of rules with many antecedents are

generated. Similar to the extended explanations based on rule traces in symbolic AI, the large rule sets extracted from neural networks offer limited or no explanation capability.

An additional example of the limitations of explanation capabilities in symbolic AI systems is described in Moore and Swartout (1989). In the field of expert systems practitioners have been linking "canned text" with rules and instead of providing the user directly with the trace of rules, the sequence of pre-prepared text elements has been used to facilitate comprehensibility. This type of user explanation is very rigid, systems always interpret questions in the same way, and there are no adequate response strategies. Although efforts have been made to take advantage of natural-language dialogues including mixed initiatives, user-models and explicitly planned explanation strategies, there is little doubt that these systems are inflexible and rigid (Andrews et al., 1995).

While the integration of an explanation capability (via rule extraction) within a trained neural network or SVM is crucial for user acceptance, such systems must avoid the problems already encountered in symbolic AI.

2.2 Transparency

The creation of a *"user explanation"* capability is the primary objective for extracting rules from neural networks and SVMs, with the provision of "transparency" of the internal states of a system a close second. Transparency means that internal states of the machine learning system are both accessible and can be interpreted unambiguously. Such transparency would allow the exploration of regions in generalisation space which may lead to erroneous or sub-optimal decisions.

Such a capability is mandatory if neural network or SVM based solutions are to be accepted into *"safety-critical"* problem domains such as air traffic control, the operation of power plants, medical surgery, etc. Rule extraction offers the potential for providing such a capability (Andrews et al. 1995).

2.3 Software Verification

If neural networks or SVMs are to be integrated in larger software systems that need to be verified, then clearly this requirement must be extended to all components, including the ANNs and SVMs. Currently, rule extraction algorithms do not allow for verification, i.e. they do not *prove* that a machine learning system behaves according to some specification. However, rule extraction algorithms provide a mechanism for either partially or completely *"decompiling"* a neural network or SVM. This is about half-way to software verification because it allows for a comparison between the extracted rules and the specification.

2.4 Improving Generalisation

If a limited or unrepresentative data set has been used in the ANN training process, it is difficult to determine *if and when generalisation fails for specific cases* even with evaluation methods such as cross-validation. By expressing learned knowledge as a set of rules, an experienced user can anticipate or predict a *generalisation failure* (Andrews et al., 1995). It may also be possible to identify regions in input space that are not represented sufficiently in the data set and need to be supplemented (Andrews et al. 1995).

2.5 Data Exploration and the Induction of Scientific Theories

This has been one of the primary objectives for rule extraction from neural networks and is essential for data mining and knowledge discovery. As Craven and Shavlik (1994) write *"a (learning) system may discover salient features in the input data whose importance was not previously recognised"*. If a neural network or SVM has learned important and possibly non-linear relationships, these relationships are encoded incomprehensibly as weight vectors, support vectors and additional parameters. Within the context of discovering new relationships, rule extraction algorithms significantly enhance the data mining capabilities of neural networks and SVMs.

3 Translucency and Rule Quality

Over the last years, a number of studies on rule extraction from support vector machines have been introduced. The research strategy in these projects is often based on this idea: develop algorithms for rule extraction based on the perception (or "view") of the underlying SVM which is either explicitly or implicitly assumed within the rule extraction technique. In the context of rule extraction from neural networks the notion of "translucency" describes the degree to which the internal representation of the ANN is accessible to the rule extraction technique (Andrews et al. 1995; Tickle et al. 1998). More broadly, a taxonomy for rule extraction from neural networks has been introduced (Andrews et al. 1995; Tickle et al. 1998) which includes five evaluation criteria: translucency, rule quality, expressive power, portability and algorithmic complexity. These evaluation criteria are now commonly used for rule extraction from SVMs.

It is important to develop new techniques for rule extraction from support vector machines, including those that are based on SVMs only and do not require any other machine learning technique. In particular support vector machines that allow the generation of structured outputs (Taskar et al. 2005) can be used to generate rule sets not unlike those extracted from neural

networks. This represents a clear advancement since user explanation is realized by an SVM and not by a technique with a different representational bias. In addition, methods for the extraction of high quality rule sets from SVMs trained on high-dimensional data are required.

The following briefly describes the first two of the five evaluation criteria for rule extraction from neural networks (Andrews et al. 1995; Tickle et al. 1998) which are then discussed in the context of rule extraction from SVMs.

3.1 The Neural Network Case

Translucency describes the degree to which the internal representation of the ANN is accessible to the rule extraction technique. At one end of the translucency spectrum are those rule extraction techniques which view the underlying ANN at the maximum level of granularity, i.e. as a set of discrete hidden and output units. Craven and Shavlik (1994) categorized such techniques as "decompositional". The basic strategy of decompositional techniques is to extract rules at the level of each individual hidden and output unit within the trained ANN. In general, decompositional rule extraction techniques incorporate some form of analysis of the weight vector and associated bias (threshold) of each unit in the trained ANN. Then, by treating each unit in the ANN as an isolated entity, decompositional techniques initially generate rules in which the antecedents and consequents are expressed in terms which are local to the unit from which they are derived. A process of aggregation is then required to transform these local rules into a composite rule base for the ANN as a whole (Tickle et al. 1998).

In contrast to the decompositional approaches, the strategy of pedagogical techniques is to view the trained ANN at the minimum possible level of granularity, i.e. as a single entity or alternatively as a "black box". The focus is on finding rules that map the ANN inputs (e.g. the attribute/value pairs from the problem domain) directly to outputs (Tickle et al. 1998). In addition to these two main categories, Andrews et al. (1995) also proposed a third category which they labelled as "eclectic" to accommodate those rule extraction techniques which incorporate elements of both the decompositional and pedagogical approaches.

A number of authors have studied the algorithmic complexity of extracting rules from feedforward neural network. Here is a brief summary of results:

- Decompositional approach: The basic process of searching for subsets of rules at the level of each (hidden and output) unit is *exponential in the number of inputs to the node.*
- Heuristics are invoked to limit the depth to which the space is explored.
- Golea (1996) showed that *extracting the minimum DNF (disjunctive normal form) expression from a trained feedforward net is hard in the worst case.*

- Furthermore, Golea (1996) showed that the Craven and Shavlik (1994) algorithm is *not* polynomial in the worst case.
- This result does not apply to single-layer networks; however, extracting the best N-of-M rule from a single-layer network is again hard.

Rule extraction from neural networks early on adopted criteria for the quality of the extracted rules. The set of criteria for evaluating rule quality includes (Andrews et al. 1995):

1. Accuracy
2. Fidelity
3. Consistency, and
4. Comprehensibility of the extracted rules

A rule set is considered to be accurate if it can correctly classify a set of previously unseen examples from the problem domain (Tickle et al. 1998). Similarly a rule set is considered to display a high level of fidelity if it can mimic the behaviour of neural network from which it was extracted by capturing all of the information represented in the ANN. An extracted rule set is deemed to be consistent if, under differing training sessions, the neural network generates rule sets which produce the same classifications of unseen examples. Finally the comprehensibility of a rule set is determined by measuring the size of the rule set (in terms of the number of rules) and the number of antecedents per rule (Tickle et al. 1998).

3.2 Translucency and Rule Quality Applied to Rule Extraction from SVMs

Most current studies on rule extraction from SVMs focus on decompositional extraction; however, learning-based approaches are also available (Barakat and Diederich 2005). The idea is simple: learn what the SVM has learned. For this purpose a data set is divided into two or more parts. The first set is used to train the SVM to completion. The second set does not include targets, the inputs are presented to the SVM and the output is obtained from the SVM. Inputs and outputs combined represent a new data set that is used for a second machine learning episode by use of a machine learning system that produces rules as output.

Hence, pedagogical rule extraction from SVMs is trivial, in particular if the data set is low-dimensional. Support vector machines have been designed to process high-dimensional input data. Typical examples are text, speech and image classification. Yet most of the studies available on rule extraction from support vector machines use benchmark data sets that include a limited number of features only. As a result, SVMs are not being used in their core application area and hence rule extraction results are not very meaningful.

It is very easy to illustrate the limitations of current studies on rule extraction from SVMs by use of an example: text classification. SVMs can

achieve good performance with very simple text representation formats such as the "bag-of words" (BOW) technique. BOW uses a document-term matrix such that rows are indexed by the documents and columns by the terms (e.g. words). SVMs allow the classification of texts of differing lengths; hence, document vectors may differ greatly in the number of elements.

A disadvantage of the BOW representation is that after successful classification, it may not be obvious *what* has been learned. For instance, an author may have a preference for certain topics and as a result, an SVM trained on an authorship identification problem in reality may perform topic detection. This problem has lead to various techniques to eliminate content from the BOW input, for instance by replacing content words with lexical tags (categories).

Given the fact that it is not at all obvious what contributes to classification in case of a BOW input representation, rule extraction from support vector machines is presented with a special opportunity. However, the number of features can be very large: e.g. all words that exist in a given natural language. While a combination of words constitutes meaning in a natural language, BOW and hence classification is based on words in isolation. This is a significant problem with regard to rule quality: The antecedents in a rule include individual words completely out of context. As the set of antecedents includes completely unrelated words, *human or semantic comprehensibility* is low.

4 An Alternative View on Rule Extraction: Information Retrieval

The introduction to explanation in Sect. 1 neglected one very important aspect: Explanation is frequently based on an interaction between two persons (e.g. a teacher and a student) or a machine and a human (e.g. in tutorial or help systems). Cawsey (1993) uses a very simple definition of explanation: "In general an explanation is something which makes some piece of knowledge clear to the hearer.... The explanation is complete when the hearer is satisfied with the reply and understands the piece of knowledge" (Cawsey, 1993, p. 1). Hence, explanation is based on an "information need" and essentially is a dialogue. As part of this dialogue, explanatory targets may change and may be refined.

Frequently, it is possible to retrieve more than one explanation (i.e. rule set) from a given SVM or neural network. In this case, it is often necessary to select the best explanation. There are two main reasons why it is possible to generate multiple explanations from or for a given machine learning system (1) Rule extraction methods include parameters that need to be initialized and the selection of certain options or values for variables results in different rule sets. (2) Rules can be expressed in different ways, e.g. a rule set with few rules that have many antecedents can be re-written as a rule set with many, simple rules, i.e. rules with few conditions. While it is generally acknowledged

that a rule set with few rules and a limited set of antecedents provides best explanation capability, it is far from obvious that this is always the case. Indeed, the *information need* of the user has to be taken into consideration and the user may *interact* with the machine learning system by use of rule extraction.

A user may have a need for multiple explanations (rules sets) because the objective is to explore the generalization space of the underlying SVM or neural network. Several rule sets, if considered in turn, may offer best transparency of the ANN or SVM. In another scenario, a user may be interested in the single-best explanation in the form of a few simple rules. In addition, the user may be interested in exploring different parameter sets which lead to different learning results and consequently to different rule sets. It is difficult to consider rule quality criteria without reference to the information need of the user.

The concept of *information need* is central to the discipline of information retrieval. The performance of an information retrieval system, e.g. an Internet search engine, is traditionally evaluated by use of "precision" and "recall". Precision is the probability that a document predicted to be genuine truly belongs to this class. In other words, a document that has been retrieved from a database truly matched the information need of the user. Recall is the probability that a genuine document is classified into this class. Less formally, high recall is given if all documents that satisfy the information need of the user are indeed retrieved from a database.

It is obviously desirable to have high recall and precision simultaneously but this is difficult to achieve in information retrieval. A trade-off exists between large recall and precision. By adjusting a parameter, e.g. by altering the cost of misclassification, recall may be increased at the cost of decreasing precision and vice versa.

The observation that multiple explanations can be extracted from a trained SVM or neural network leads to the application of information retrieval concepts to rule extraction (see the case study below). In the context of rule extraction from either support vector machines or neural networks, high precision represents the scenario that the rule sets extracted are relevant to the user, i.e. match his or her information need. Recall refers to the question "how many relevant explanations that can be extracted from the SVM or neural network are indeed being extracted?" Is it possible to generate all possible explanations by way of rule extraction that match the information need of the user?

The ideas outlined above may lead to the application of additional performance measurements that are commonly used in information retrieval and that are based on precision and recall. The view that a single explanation is to be extracted from an SVM or neural network is a simplification. It may be desirably to extend the notion of "rule quality" to include assessments of multiple explanations in relation to the information need of a user.

5 A Case Study

To illustrate the problem (and opportunity) faced by rule extraction from SVMs when applied to text classification, support vector machines are used to classify business news articles from the Persian Gulf with regard to emotional content. A total of 914 news articles are used for this experiment.[1] In addition, experiments on authorship attribution (identifying the author of a text) and topic classification are performed.

The pre-processing includes two parts: text extraction and feature selection. Text extraction is performed by lexical analysis to strip all non-word annotations and to convert the text into a list of words or tokens. This step can be summarised as follows: (1) upper case letters are converted to lower case, (2) all words containing non-letter characters are removed including hyphenated words and words with an underscore, (3) all punctuations are replaced with space characters to be treated as token delimiters, (4) author identities are extracted, (5) the texts are converted to a "bag-of words" (BOW) representation.

In addition, all class identifiers are removed from the articles in the BOW format before they are used to generate a fixed vocabulary: author names and words used for topic and emotion identification are removed (see the section below on the clustering process used to generate targets for supervised machine learning). After the text extraction process, a fixed length vocabulary is built from the set of all extracted news articles through a feature selection process. Firstly, stopword removal and stemming[2] are performed on each extracted text. Secondly, document frequency thresholding is used to reduce further the feature vector space. Words occurring once only are removed.

After the vocabulary generation process, for each class (four authors, emotions and topics), the extracted texts are (1) labelled with the class and (2) mapped to an SVM data file in which each line represents a news article. Each row includes a label that indicates whether the article belongs to the target class or not.

Pre-processing of the text samples, including elimination of frequent words (using an edited list of the 6,500 most frequent words in English) led to the development of lists of words that are low frequent, but included some words with topical or emotional content which are common. Simple clustering techniques are used to extract topic and emotion information from texts to perform supervised learning.

[1] This is joint work with Insu Song, Aqeel Al Ajmi, Jihan Zhu, Imran Fanaswala and Mark Pedersen.

[2] Stopword removal refers to the elimination of function words such as articles (the, a) Stemming identifies the root of a word, e.g. "goes" will be converted to "go".

The following is a description of the algorithm used to identify topical and emotive information:

For all business news articles

1. Generate a ranked list of n words that are not in a stoplist (comprising an edited list of the 6,500 most spoken words in English)
2. Apply cluster analysis to the ranked word lists extracted from the documents
3. Identify words that are high-frequent in clusters

The method in step 2 is described by Chiu et al. (2001) who proposed a conceptual or model-based approach to hierarchical clustering. The method includes a two-step strategy to determine the number of clusters. The model associated with a cluster covers both numerical and categorical attributes and constitutes a mixture of Gaussian and multinomial models. The distance between two clusters is defined as a decrease in log-likelihood caused by merging of the two clusters under consideration. The process continues until a stopping criterion is met. As such, determination of the best number of clusters is automatic (Berkhin, 2002).

The emotion categories extracted by this process are "boom", "confident", "regret" and "demand". The topic categories are "asian economy", "oil price", "stock" and "gas". As in previous studies, authorship attribution succeeds at a very high level. All four attempts to identify the author of the texts are successful (Level-one-out cross-validation estimates of the performance: error $<2\%$, precision 100% and recall 70–97%). Topic detection performance is lower and on par with the emotion classification results. Two of the topic detection learning results are relatively poor as are two of the attempts to discover the emotion express in the text.

Pedagogical rule extraction from SVMs as outlined in Barakat and Diederich (2005) is applied to the trained SVMs (one authorship attribution as well as one topic detection and emotion classification problem). The procedure for rule extraction is as follows:

1. Divide data in two or more sets
2. Train SVM on a subset of data A
3. Get SVM predictions on subset B
4. Combine inputs from subset B with SVM predictions
5. Train a symbolic machine learning system on the new data set
6. Obtain rules from the symbolic machine learning system (in this case a decision tree learner and a classification and regression tree)

Since the decision tree learning system cannot efficiently deal with high-dimensional input spaces, the first 200 features are used for See 5 learning only. The following rules (Table 1) were extracted in one run from an SVM

Table 1. Rules extracted from an SVM trained on an authorship identification task

Rule 1:	interest <= 0 percent > 0.079 product <= 0.086 -> Target author
Rule 2:	us <= 0 price <= 0.027 compani <= 0.031 set > 0.061 -> Target author
Rule 3:	market <= 0.075 number > 0.1 -> Target author
Rule 4:	percent <= 0.079 product <= 0.086 number <= 0.1 set <= 0.061 -> Other author
Rule 5:	us > 0 number <= 0.1 -> Other author
Rule 6:	price > 0.027 -> Other author
Rule 7:	market > 0.075 -> Other author
Default class:	Other author

trained on an authorship identification problem. Please note that in this case, the SVM classifications for inputs in the training set A are used.

The training error of the decision tree learning method is low: 2.3%. There are rules for the positive and negative classes and rule quality is high: seven rules and 17 antecedents in total (of the 200 possible features, only nine occur in the rules). The rules include content words (or rather word stems) only since pre-processing eliminated all function words. Word frequency distribution over function words is relevant for authorship attribution; nevertheless, even without function words SVM and decision tree learning succeeds.

The following rules were extracted from the topic-detection SVM (Table 2). Again, note that SVM classifications for inputs of the training set A are used as targets for decision tree learning.

The See 5 learning error is 5.0% and rule quality is obviously slightly reduced. Rule 6 is particularly problematic, a point which will be discussed in detail further below.

Finally, rules were extracted from an SVM trained on an emotion classification problem (Table 3). The learning result for this SVM is acceptable (leave-one-out cross-validation result: error 8.6%, precision 94.9%, recall 70%).

Table 2. Rules extracted from an SVM trained on a topic detection problem

Rule 1:	exchang > 0.089 valu <= 0.024 -> Target topic
Rule 2:	share > 0.083 -> Target topic
Rule 3:	investor > 0.053 advanc > 0.043 -> Target topic
Rule 4:	market > 0.046 fed <= 0.076 -> Target topic
Rule 5:	market <= 0.046 share <= 0.083 exchang <= 0.089 advanc <= 0.043 -> Other topic
Rule 6:	oil <= 0.318 investor <= 0.033 share <= 0.083 world < = 0.077 foreign <= 0.091 jordan <= 0.053 exchang <= 0.089 long <= 0.068 export <= 0.097 posit <= 0.053 hous <= 0.058 -> Other topic
Rule 7:	fed > 0.076 -> Other topic
Rule 8:	german > 0.057 japan <= 0.032 -> Other topic
Rule 9:	share <= 0.083 properti > 0.183 -> Other topic
Rule 10:	dollar > 0.336 -> Other topic
Rule 11:	sterl > 0.078 -> Other topic
Default class:	Other topic

In contrast to the two cases above, SVM classifications for inputs of the test set B are used as target for decision tree learning.

The test set includes 120 cases and the See 5 learning error is 1.7%. Tenfold cross-validation reveals a test error of 13.3%, however, practically all positive examples are misclassified. Clearly, there are an insufficient number of positive examples in this data set (8 out of 120).

It is possible to use sub-sampling techniques or boosting trials to elicit more words (antecedents in rules) in order to identify relevant features or to clarify the classification task. Ten boosting trials lead to rule sets that confirm the result above on the one hand but also identify additional features (Table 4). The boosted decision tree classifier correctly learns all cases.

Table 3. Rules extracted from an SVM trained on an emotion classification problem

```
Rule 1:         suppli > 0
                estat > 0.032
                -> Target emotion
Rule 2:         close > 0.02
                estat > 0.032
                -> Target emotion
Rule 3:         estat <= 0.032
                -> Other emotion
Rule 4:         estat > 0.032
                -> Other emotion
Default class: Other emotion
```

Table 4. Rules extracted by use of a tenfold boosting run from an SVM trained an emotion classification problem

```
Rule 0/1:       suppli > 0
                estat > 0.032
                -> Target emotion
Rule 0/2:       close > 0.02
                estat > 0.032
                -> Target emotion
Rule 0/3:       estat <= 0.032
                -> Other emotion
Rule 0/4:       estat > 0.032
                -> Other emotion
Default class: Other emotion
```

```
Rule 1/1:       war > 0.022
                -> Target emotion
Rule 1/2:       war <= 0.022
                -> Other emotion
Default class: Other emotion
```

Table 4. (Continued)

Rule 2/1:	al > 0.024 high > 0.028 chang <= 0 -> Target emotion
Rule 2/2:	chang > 0 -> Other emotion
Rule 2/3:	high <= 0.028 -> Other emotion
Rule 2/4:	al <= 0.024 -> Other emotion
Default class:	Other emotion
Rule 3/1:	oil > 0.222 -> Target emotion
Rule 3/2:	estat > 0.046 -> Target emotion
Rule 3/3:	oil <= 0.222 estat <= 0.046 -> Other emotion
Default class:	Other emotion
Rule 4/1:	uae > 0.193 -> Target emotion
Rule 4/2:	higher > 0.154 -> Target emotion
Rule 4/3:	higher <= 0.154 uae <= 0.193 -> Other emotion
Default class:	Other emotion
Rule 5/1:	hous > 0 -> Target emotion
Rule 5/2:	hous <= 0 -> Other emotion
Default class:	Other emotion
Rule 6/1:	compani <= 0.05 gulf <= 0 estat > 0.032 -> Target emotion
Rule 6/2:	suppli > 0.058 -> Target emotion
Rule 6/3:	suppli <= 0.058 estat <= 0.032 -> Other emotion

(*continued*)

Table 4. (Continued)

```
Rule 6/4:      compani > 0.05
               -> Other emotion
Rule 6/5:      gulf > 0
               -> Other emotion
Default class: Other emotion
```

```
Rule 7/1:      dubai <= 0.087
                 monei > 0.05
               ->  Target emotion
Rule 7/2:      dubai > 0.087
               -> Target emotion
Rule 7/3:      dubai <= 0.087
               monei <= 0.05
               -> Other emotion
Default class: Other emotion
```

```
Rule 8/1:      declin > 0.09
               -> Target emotion
Rule 8/2:      iraq > 0.028
               -> Target emotion
Rule 8/3:      iraq <= 0.028
               declin <= 0.09
               -> Other emotion
Default class: Other emotion
```

```
Rule 9/1:      project > 0.116
               -> Target emotion
Rule 9/2:      real > 0.177
               -> Target emotion
Rule 9/3:      real <= 0.177
               project <= 0.116
               -> Other emotion
Default class: Other emotion
```

It is obvious that different machine learning techniques that "learn what the SVM has learned" may produce different results (the value of extracting a range of different rule sets from an SVM has been outlined above). If a classification and regression tree is used to generate the explanation, two simple rules are generated (Table 5).

The rule sets above allow a number of interesting observations. In the case of the "authorship attribution" rule set (Table 1), it is not at all clear what contributes to the identification of this author. All documents are business news articles, hence the question is: does this author focus on a particular

Table 5. Rules extracted from an SVM by use of a classification and regression tree. The SVM has been trained an emotion classification problem

```
Rule 1: valu < 0.047
        invest < 0.275
        -> Other emotion
        invest >= 0.275
        -> Target emotion
Rule 2: valu >= 0.047
        currenc < 0.016
        -> Other emotion
        currenc >= 0.016
        -> Target emotion
```

topic (e.g. the US stock market) or do the rules capture relevant features of the author's style?

The rule sets obtained from the SVM trained on the emotion classification problem are even more intriguing: Rule quality is very high, rule extraction is consistent and the learning results are good. However, *none of the antecedents express emotion in any way at all!* Even though all SVM and decision tree learning results are acceptable (with exception of the "extreme" k-fold cross-validation), *the rules appear not to be linked to the task at hand*: classify documents into categories that express the emotion of a text. Since it is one of the objectives of rule extraction to explain "how" classification is realized by an SVM, the question must be asked to what extend the above rule sets help to provide an answer.

The experiments outlined above invite a number of objections that should be discussed in the context of evaluating the rules. First of all, the decision tree learner was trained by use of the 200 most frequent features (word stems) only, the SVM considers up to 6,484 attributes. It is possible that the vast majority of features that are not input to the decision tree learner are relevant for SVM learning (the feature ranked 200 in the corpus has an absolute frequency of 291; generally, word stems that occur twice or more in the entire data set are being considered for SVM learning). Joachims (1998) established that the majority of words with low frequency in a corpus do contribute to a text classification task. Hence, it is possible that the example rule sets above do not completely capture what the SVM has learned.

Since it is difficult to train a decision tree learner or a classification and regression tree on a high-dimensional data set, it may be argued that this is a case for a "decompositional" extraction method that does not rely on a non-SVM learning technique so obviously insufficient for high-dimensional problems. It is unlikely; however, that such an approach offers a solution to the problem. Obviously, individual words or word stems taken out of context cannot provide a human comprehensible explanation.

The four rules in Table 3 all include the antecedent "estat". This is a word stem generated from the word "estate" which occurs 362 times in the corpus. Practically all of the occurrences are in the context of "real estate", and since this is a corpus including business news articles from the Persian Gulf, "real estate" refers to the current construction boom in Dubai. Table 3 does not include any reference to "real" (as in "real estate" or "Dubai"). The boosting trials in Table 4, however, extract the words "real" and "Dubai". Similarly, the word stem "suppli" originated from "supplies" or "suppliers" and is used in the context of "oil supplies".

In a further experiment, a neural network was trained on the data used for rule extraction (Tables 3 and 4). The neural network achieved more than 90% accuracy in various configurations. For instance, a feedforward neural network with a 200 unit input layer, a first hidden layer of 7 and a second hidden layer with nine units as well as a one unit output layer achieves an accuracy of 90.5%. After completion of the training, sensitivity analysis was performed on the neural network to obtain information on the relevance of input features. Sensitivity analysis ranks all inputs to the neural network (in this case 200) according to the relevance of the feature for the classification task. In various runs with different neural network architectures, "real" and "estat" were both ranked by sensitivity analysis among the top 5 input features to the neural network. "suppli" was also ranked as a top feature by sensitivity analysis. These results confirm the relevance of the features in Table 3, the rules in Tables 3 and 4 as well as the interpretation provided above.

Finally, the classification and regression tree rules in Table 5 use the word stem "valu" which occurs 355 times in the corpus. The original word forms are "value", "valuation", and "devaluation" and so on. "invest" is one of the most frequent word stems (1,543 occurrences) similar to "currenc" (974 occurrences). Obviously, the original words appear in various contexts.

The rules in Tables 3–5 point only indirectly to the criteria the SVM utilises for classification. Again, it is important to emphasise that the SVM in question has been trained on an emotion classification task, yet, rule extraction does not reveal any word that is linked to affect or emotion. Sensitivity analysis after neural network training on the same data, however, extracts some words that do have emotional content such as "profit". There is some indication that in reality, the SVM performs a topic classification task. The rule extraction process could be considered a success since a possible confusion between a topic and emotion classification task has been discovered. Yet again, in this particular case the SVM does perform emotion classification to some extent as indicated by the ROC curve in Fig. 1 with an "area under the curve" AUC of .86. The extracted rules should be used in full view of the learning result of the SVM in order to explain what the support vector machine has learned.

Is it possible to say that SVM emotion classification by use of a "bag-of-words" representation has failed because no emotion word occurs in the extracted rules? No, it is not possible to say this conclusively because the

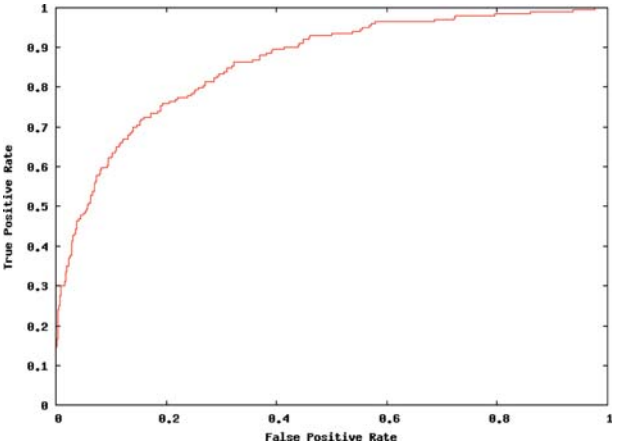

Fig. 1. The ROC curve for the emotion classification SVM. Please note that this ROC curve has been obtained from an SVM trained on a slightly larger data set (914 patterns). The SVM used for rule extraction has been trained on 734 examples

SVM has been trained on a high-dimensional data set while the decision tree learner used for extraction utilizes 200 features only. Also, the boosting runs in Table 4 generated rules including words like "war", "iraq", "declin" (for decline) and potentially several others which do have emotional content at the point in time this study was performed. Nevertheless, there is evidence that input dimensionality is crucially important for rule extraction from SVMs!

In order to enhance the explanatory value of the rules extracted from the SVMs, it may be beneficial to train an additional SVM on "bigram frequencies" and to extract rules from this SVM. "Bigrams" are sequences or combinations of words that appear in sentences. For instance, the sentence "Real estate in Dubai is expensive" includes "real estate" and "in Dubai" as bigrams. If stopwords such as "in" and "is" are eliminated, bigrams such as "real estate" and "Dubai expensive" would be generated. As part of the pre-processing of documents for SVM learning, the frequency of occurrence of bigrams would be calculated and would be utilised in the SVM data set. For instance, bigrams such as "real estate" would be attributes in the SVM data set and the normalised frequency of occurrence would be the value of the feature.

Rule extraction from SVMs would then generate rules including antecedents such as "real estate", "in Dubai" or "Dubai expensive". Due to the added context, rules with these antecedents would be much more comprehensible. Even if the SVMs trained on bigrams frequencies do not perform as well as those trained on a simple "bag-of-words" representation, the bigram rules can be compared to those extracted from an SVM trained on "bag-of-words". It is common in artificial intelligence systems to have an independent explanation

system, i.e. a system separate from the core inference machine. In a similar sense, the rules extracted from the bigram SVM can be used for explanatory purposes only.

The observations above lead to a new rule quality criterion: *semantic comprehensibility*. In the cases outlined above, rule quality as originally formulated (Tickle et al. 1998) is high due to the limited number of antecedents and rules, yet *comprehensibility* for the user is low. Therefore, it is necessary to introduce a new rule quality criterion, semantic comprehensibility.

6 A Classification System for Rule Extraction from SVMs

Rule extraction from support vector machines requires evaluation criteria that emphasize data (Table 6). SVMs have demonstrated very good performance when trained on data sets with high-dimensional inputs; significant application areas are image classification (including face recognition), bioinformatics and text classification. At this point in time, many Internet search engines use support vector machines.

In light of the discussions above, the following dimensions are proposed:

1. Translucency: This dimension as originally proposed in Andrews et al. (1995) and Tickle et al. (1998) continues to be useful, even with the

Table 6. A classification system for rule extraction from SVMs

Attribute	Type	From	To
Translucency	Continuous	Decompositional	Pedagogical
Data	Continuous	Low-dimensional	High-dimensional
Expressiveness	Discrete	Boolean	First-order predicate logic
Rule quality			
Number of rules	Continuous	1	No upper limit
Number of antecedents	Continuous	1	No upper limit
Semantic comprehensibility	Discrete	Yes	No
Fidelity	Continuous	0%	100%
ROC Fidelity	Continuous	Low	High
Accuracy of rules	Continuous	0%	100%
Precision of rules	Continuous	0%	100%
Recall of rules	Continuous	0%	100%
Complexity	Continuous	Linear	Exponential
Non-SVM extraction	Discrete	Yes	No

introduction of rule extraction techniques for high-dimensional data sets. However, there are important questions with regard to the learning-based or pedagogical approach: Many of the rule-based learners have very different representational biases compared to support vector machines. In addition, many are less suitable for high-dimensional inputs, again in comparison to SVMs. These limitations have been discussed already by use of the case study.

2. Data: From low to high-dimensional input space. Many data mining practitioners would probably agree with the view that it is possible to *engineer* a neural network with similar learning performance to an SVM for low-dimensional data sets. As a matter of fact, SVMs are not necessarily the method of choice for these data sets, there are many alternatives. This includes those cases with relationships between attributes that are best expressed as a decision tree. Hence, the view here is that SVMs are the primary choice for high-dimensional inputs and rule extraction techniques should work in these cases.
3. Expressiveness of the extracted rules. Rule extraction from neural networks has previously almost exclusively been used to generate propositional rule sets (Hayward et al. 2000). While this is sufficient for many applications where rule sets can be effectively used, it is clearly desirable to provide a more general explanation capability. Hayward et al. (2000) describe an approach to representing a neural network as a PROLOG logic program, where the activation values of hidden and output units are equated with the truth value of predicates. The technique addresses several issues. Hayward et al. (2000) describe a process whereby Boolean formulae are translated into a first-order representation consisting of predicates, rules and facts. This is a field that is largely unexplored in the context of rule extraction from support vector machines; however, it is conceivable that SVMs with structured output (Taskar et al. 2005) will lead to complex rule sets and languages beyond and above propositional logic. Also, please see the chapter by Torrey et al. in this volume.
4. Rule Quality: This category includes accuracy, fidelity and comprehensibility. "Semantic comprehensibility" is given if minimal rules sets with concise rules are extracted from SVMs trained on high-dimensional data. Fidelity may be extended to "ROC fidelity", e.g. if and when the SVM and the rule set exhibit the same classification behaviour with modified cost functions.

   It is crucial to consider the sub-category "semantic comprehensibility" and the case study above is designed to outline some of the relevant issues. Given high-dimensional data sets, features in isolation have limited or no explanation capability. It has been proposed earlier to extract multiple rule sets from SVMs to explore the full feature set that contributes to a classification. In many ways, it is the user who decides which rule set has value and hence, the notion of "information need" has been adopted from information retrieval. Semantic comprehensibility

is not formally defined here and is not proposed as a quantitative measure. At this stage, "semantic comprehensibility" refers to the ultimate goal to extract user-comprehensible rule sets from *any* SVM.

5. Complexity of the extraction. To date, there has been no systematic study on the algorithmic complexity of rule extraction from SVMs. The results that have been obtained for rule extraction from neural networks (some have been summarised above) are not applicable because they rely on (1) the structure of the neural network (single or multi-layer) and (2) properties of a learning algorithm such as backpropagation. Many of the current approaches for rule extraction from SVMs include heuristics and/or machine learning or statistical techniques that are interchangeable. For instance, the learning-based rule extraction from SVM technique used in the case study above uses either a decision tree learner or a classification and regression tree. Núñez et al. (2002) use clustering techniques with the aim to identify regions in decision space that can be translated to rules. Opportunities and limitations of the rule extraction from support vector machines enterprise are yet to be fully explored.
6. Non-SVM extraction: It is important to develop new techniques for rule extraction from support vector machines, including those that are solely based on SVMs and do not require any other machine learning technique. In particular, support vector machines that allow the generation of structured outputs (Taskar et al. 2005) can be used to generate rule sets not unlike those extracted from neural networks. This represents a clear advancement since user explanation is realized by an SVM and not by a technique with a different representational bias. In addition, methods for the extraction of high quality rule sets from SVMs trained on high-dimensional data are required.

By way of example, it is easily possible to apply this classification system to the simple case study provided above, the emotion classification problem.

Table 7 clearly demonstrates the limitations of the algorithm used in the case study. Rules lack expressiveness, the extraction process does not really consider the dimensionality of the data set, and while rule quality is quite good, the algorithmic complexity of the extraction process is not satisfactory.

7 Conclusions and Future Challenges

It is obviously not possible to discuss all aspects of rule extraction from support vector machines in this brief introduction. There is one area in particular that would deserve a fuller consideration. This is the use of committee

Table 7. The classification of the case study in Sect. 5

Translucency	Pedagogical
Data	Low-dimensional (SVM is trained on high-dimensional data)
Expressiveness	Boolean
Rule quality	
Number of rules	2–11
Number of antecedents	5–28
Semantic comprehensibility	No
Fidelity	Not tested
ROC Fidelity	Not tested
Accuracy of rules	90%
Precision of rules	66%
Recall of rules	90%
Complexity[3]	$O(mn \log n) + O(n(\log n)^2)$
Non-SVM extraction	No

machines or ensemble learning approaches.[4] Obviously, it is more difficult to extract comprehensive rules from a set of support vector machines or an ensemble of machine learning techniques of different type. The requirements for comprehensibility are even harder to meet if more than one classifier is involved.

A first attempt has been made in a project to predict the return of stocks in the US market.[5] An SVM was trained to accept input from various machine learning techniques to predict the next day return of shares. The machine learning methods includes various types of neural networks, support vector machines as well as an implementation of Ripper. The target for the SVM-based committee machine is the next day return. Rules were then extracted which took the following form:

> IF the value of the prediction of C4.5 is equal to -1 THEN the SVM committee machine classifies the sample as -1, Otherwise, it classifies the case as $+1$.

Given the low-dimensionality of the input, SVM learning (i.e. the committee machine) did not significantly improve the overall results and the SVM tends to agree with the decision tree learner C4.5

[3] Decision tree learning complexity for C4.5 according to Witten and Frank (1999, p. 168). n is the number of samples and m the number of attributes. See 5 has been used in this study.

[4] Thank you to Alan Tickle for suggesting the importance of committee machines.

[5] This work was performed by Hanan Tayeb, Shahrazad Mohammed, Ghasaq Yousif and Shrouq Hasan as part of a final year undergraduate project, Department of Computer Science, American University of Sharjah, Spring 2007.

The objective of this introductory chapter is to outline a number of research issues, some of which are addressed in the following chapters. The ultimate goal, however, is to achieve what rule extraction from neural network undoubtedly has achieved, and this is to propose a set of techniques suitable for data mining and commercial applications. The following chapters include new algorithms for rule extraction as well as applications in a variety of domains. This includes financial applications as well as speech recognition. There is no doubt that the current research on developing new kernel methods to increase the accuracy of classification and regression must be complemented by a set of techniques that allow user explanation at a very high level.

8 Acknowledgements

Thank you to David Martens, Lisa Torrey, Shlomo Geva and Alan Tickle for comments on earlier versions of this introduction.

References

Andrews R, Diederich J, Tickle AB (1995) A Survey and Critique of Techniques For Extracting Rules From Trained Artificial Neural Networks, Knowledge Based Systems, 8, pp. 373–389

Barakat N, Diederich J (2005) Eclectic rule extraction from support vector machines. International Journal of Computational Intelligence, 2(1), 59–62, 2005

Berkhin P (2002) Survey of Clustering Data Mining Techniques. Accrue Software, San Jose, California

Cawsey A (1993) Explanation and Interaction. The Computer Generation of Explanatory Dialogues. The MIT Press Cambridge London

Chiu T, Fang D, Chen J, Wang Y, Jeris C (2001) A Robust and Scalable Clustering Algorithm for Mixed Type Attributes in Large Database Environment. Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 263

Craven M, Shavlik J (1994) Using sampling and queries to extract rules from trained neural networks. In Proceedings of the 11th International Conference on Machine Learning, 1994, 37–45

Davis R, Buchanan BG, Shortliffe E (1977) Production rules as a representation for a knowledge-based consultation program'. Artificial Intelligence 8:1 15–45

Gallant S (1988) Connectionist expert systems. Communications of the ACM. 13:2 152–169

Gilbert N (1989) Explanation and dialogue. The Knowledge Engineering Review. 4:3 235–247

Golea, M (1996) "On the complexity of rule extraction from neural networks and network querying". Proceedings of the Rule Extraction from Trained Artificial Neural Networks Workshop, Society for the Study of Artificial Intelligence and Simulation of Behavior Workshop Series (AISB'96) University of Sussex, Brighton, UK, 51–59

Hayward R, Nayak R, Diederich J (2000) Using Predicates to Explain Networks. In: ECAI-2000 Workshop: "Foundations of Connectionist-Symbolic Integration: Representation, Paradigms and Algorithms. Berlin, Germany

Joachims T (1998) Text Categorization with Support Vector Machines: Learning With Many Relevant Features. In: ECML-98, 10th European Conference on Machine Learning, Heidelberg, Germany, 137–142

Moore, JD, Swartout WR (1989) A Reactive Approach to Explanation. In: IJCAI-89 International Joint Conference on Artificial Intelligence, 1504–1510

Núñez H, Angulo C, Catala A (2002) Rule extraction from support vector machines. In: ECAI-92 Proceedings of European Symposium on Artificial Neural Networks, 107–112

Taskar B, Chatalbashev V, Koller D, Guestrin C (2005) Learning structured prediction models: A large margin approach. In: ICML 2005, Proceedings of the 22nd International Conference on Machine Learning

Thagard PR (1978) The best Explanation: Criteria for Theory Choice. The Journal of Philsosophy 75:2, 76–92

Thagard P, Litt A (forthcoming). Models of scientific explanation. In R. Sun (ed.), The Cambridge handbook of computational cognitive modeling. Cambridge: Cambridge University Press

Tickle A, Andrews R, Golea M, Diederich J (1998) The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural network. IEEE Transactions on Neural Networks 9:6, 1057–1068

Witten IH, Frank E (1999) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Paperback 162–164