Automatic Acquisition of Symbolic Knowledge from Subsymbolic Neural Networks

Alfred Ultsch Heng Li
FG Informatik, Philipps University of Marburg
Hans Meerwein Str. (Lahnberge)
35032 Marburg/Lahn, Germany

Abstract

Knowledge acquisition is a bottleneck in AI applications. Neural learning is a new perspective in knowledge acquisition. In our approach we have extended Kohonen's self-organizing feature maps (SOFM) by the U-matrix method for the discovery of structures resp. classes. We have developed a machine learning algorithm, called SIG*, which automated extracts rules out of SOFM which are trained to classify high-dimensional data. SIG* selects significant attributes and constructs appropriate conditions for them in order to characterize each class. And SIG* generates also differentiating rules, which distinguish classes from each other. The algorithm has been tested on many different data sets with promising results. The framework of using SIG* integrated in a system which automated acquires knowledge from learned SOFM is also presented.

1. Introduction

Knowledge acquisition is often a bottleneck in AI applications. Many expert systems use knowledge in symbolic form (e.g. rules, frames, etc.). For human experts it is, however, difficult to formulate their knowledge in these formalisms. Different approaches to the problem of knowledge acquisition have been proposed, for instance interviews with experts by knowledge engineers etc. These approaches concentrate often on how to interact with the experts in order to get a formulation of their knowledge in symbolic form. Here we follow a different approach: experts have gained their expertise by experiences, i.e. by dealing with cases. In order to get the experts' knowledge into an expert system we propose to process the case data in the attempt to learn the particularities of the domains. In this paper we use artificial neural networks (ANN) for the first step of processing the data. ANN with unsupervised learning can adapt to structures inherent in a data set, i.e. the internal structure of ANN reflects structural features in the data [Ultsch/92]. Suitable ANN exhibit the property to produce their structure during learning by the integration (overlay) of many case data. This is often termed as processing "subsymbolic" data.

Kohonen's self-organizing feature maps (SOFM) [Kohonen /89] have the property that the neighbourhood among the training data, perheps in a high dimensional space, is reflected in the neighbourhood of the units on the generated feature map, practically in a 1, 2, or 3 dimensional space. We can make use of this property of SOFM to discover structures in high dimensional data and map them into a lower dimensional space. For SOFM we have developed a method, called U-matrix method (UMM), to detect and display the structures learned from the data [Ultsch/90]. Using the UMM a trained feature map is transformed into a landscape with "hills" or "walls" seperating different regions where cases are located

[Ultsch/91a]. All cases that lay in a common basin are considered to have a strong similarity i.e. have some common structural properties. With the algorithm presented in the sequel we attempt to extract a symbolic description of the similarities from the trained SOFM, i.e. to come to a symbolic general description of the cases.

An inductive machine learning algorithm called SIG* [Ultsch/91a] takes the training data with the classification detected through the learned SOFM as input, generates rules for characterizing and differentiating the classes of the data. The work reported here is a part of the research project of knowledge processing in neural architecture. In this project we have developed a system, called WINA, which uses SIG* as a knowledge acquisition tool for a diagnosis expert system while using SOFM as a neural classifier [Ultsch/92].

In section 2 the system WINA is briefly depicted. The idea of SIG* and the way SIG* works is described with an example in section 3. Finally a summary of applications and conclusions gives an overview of this work, and suggests the future work on SIG*.

2. Overview of WINA

As depicted in Figure 1 the system WINA consists of five major modules:
¥ neural classifier
¥ neural associative memory
¥ analysing tools
¥ rule extraction
¥ inference

In WINA the raw data are firstly processed such that they can be used to train Kohonen's self-organizing feature maps (SOFM). After learning of SOFM we have the neighbourhood structure among the training data implicit on SOFM. Using analysing tools, in particular the U-Matrix method [Ultsch/91a] , the neighbourhood structure on learned SOFM can be visually recognized. The information about the structure will be stored associatively. On the other hand the training data are transfered to rule extraction. SIG* takes the training data with the classification detected through SOFM as input and generates symbolic rules. The extracted rules, the information in the neural classifier and associative memory as well and the experts' rules in addition are employed in inference.

3. Rule Generation with SIG*

SIG* has been developed in the context of medical applications [Ultsch 91a]. In this domain other rule-generating algorithms such as ID3 [Quinlan/83], for example, fail to produce suiting rules. SIG* takes a data set in the space Rn that has been classified by SOFM/UMM as input and produces descriptions of the classes in the form of decision rules. For each class an essential rule, called characterizing rule, is generated, which describes that class. Additional rules that distinguish between different classes are also generated. These are called differenciating rules. This models the typical differential-diagnosing approach of medical experts, but is a very common approach in other domains as well. The generated rules by SIG*,

Figure 1: Overview of System WINA

in particular, take the significance of the different structural properties of the classes into account. If only a few properties account for most of the cases of a class, the rules are kept very simple.

Two central problems are addressed by the SIG* algorithm:

1. how to decide which attributes of the data are significant so as to characterize each class,

2. how to formulize apt conditions for each selected significant attribute.

In order to solve the first problem, each attribute of a class is associated with a "significance value". The significance value can be obtained, for example, by means of statistical measures. For the second problem we can make use of the distribution properties of the attributes of a class. In the following we use an example to describe the SIG* algorithm. The complete and formal description can be found in [Ultsch/91a].

3.1. Selecting Significant Attributes for a Class

As an example, we assume a data set of case-vectors with five attributes Attr1, Attr2, Attr3, Attr4, Attr5. Let SOFM/UMM distinguish in the example four classes Cl1, Cl2, Cl3, Cl4. Let SVij denote the significance value of Attri in class Cj . The matrix SM=(SVij)5x4 we call "significance matrix". For our example the significance matrix may be given as follows:

SM Cl1 Cl2 Cl3 Cl4
Attr1 1.5 4 6* 3.1
Attr2 3.1 3.2 20* 6.4
Attr3 5 7.4 1.8 9.5*
Attr4 6 8.3* 5.7 2.7
Attr5 8 9.5* 6.2 7.3

In this matrix the largest value in each row is marked with an asterisk (*).

In order to detect the attributes that are most characteristic for the description of a class, the significance values of the attributes are normalized in percentage of the total sum of significance values of a class. Then these normalized values are ordered in decreasing order. For Cl1 and Cl3, for example, these ordered attributes are:

percentual
significance
Cl1
Cumulative

Attr5
33.89%
33.89%

Attr4
25.42%
59.31%

Attr3
21.19%
80.50%

Attr2
13.14%
93.64%

Attr1
6.36%
100.00%

percentual significance
Cl3
Cumulative

Attr2 *
50.38%
50.38%

Attr5
15.62%
66.00%

Attr1 *
15.11%
81.11%

Attr4
14.36%
95.47%

Attr3
4.53%
100.00%

As significant attributes for the description of a class, the attributes with the largest significance value in the ordered sequence are taken until the cumulative percentage equals or exceeds a given threshold value. For a threshold value of 50% in the above example Attr5 and Attr4 would be selected for Class Cl1. For Cl3 only Attr2 would be considered. For this class there are attributes, however, that have been marked with an asterisk (see above): Attr2 and Attr1. If there are any marked attributes, that are not considered so far, as in our example Attr1, they are also considered for a sensible description of the given class. So the descriptive attributes for our examples would be:
for Cl1: Attr5, Attr4 and for Cl3: Attr2 and Attr1.

The same algorithm is performed for all classes and all attributes and gives for each class the set of significant attributes to be used in a meanigful but not over detailed description of the class. If an attribute is exceedingly more significant than all others, (consider for example Attr2 for Cl3) only very few attributes are selected. On the other hand, if almost all attributes posses the same significance considerably more attributes are taken into account. The addition of all asterisked attribues assures, that those attributes are considered for which the given class is the most significant.

3.2. Constructing Conditions for the Significant Attributes of a Class

A class is described by a number of conditions about the attributes selected by the algorithm described above. If these conditions are too strong, many cases may not be correctly diagnosed. If the conditions are too soft, cases that do not belong to a certain class are erroneously subsumed under that class. The main problem is to estimate correctly the distributions of the attributes of a class. If no assumption on the distribution is made, the minimum and maximum of all those vectors that belong, according to SOFM/UMM, to a certain class may be taken as the limits of the attribute value. In this case a condition of the i-th attribute in the j-th class can look like
attributeij IN [minij , maxij] .
But this kind of formulization of conditions likely results in an erroneous subsumption .

If a normal distribution is assumed for a certain attribute, we know from statistics, that 95% of the attribute values are captured in the limits [meanij -2*dev ,meanij +2*dev] , where dev is the value of the standard deviation of the attribute. For other assumptions about the distribution, two parameters low and hi may be given in SIG*. For this case the conditions generated are as follows:
attributeij IN
[meanij + low * dev, meanij + hi * dev].

3.3. Characterizing Rules and Differentiating Rules

The algorithm described in 3.1. and 3.2. produces the essential description of a class. If the intersection of such descriptions of two classes A and B is nonempty, i.e. a case may belong to both classes, a finer description of the borderline between the two overlapping classes is necessary. To the characterizing rule of each class a condition is added that is tested by a differentiating rule. A rule that differentiates between the classes A and B is generated by an analog algorithm as for the characterizing rules. As significance values however, they may be measured between the particular classes A and B. The conditions are typically set stronger in the case of characterizing rules. To compensate this the conditions of the differentiating rules are connected by a logical OR.

4. Applications and Conclusion

We have tested the algorithm on many data sets from different domains. Up to now the results have been very promising. In some cases knowledge that has not been known to us, but was verified by the domain experts, has been extracted. In most cases the performance of the generated rules ranged in the 80 to 90 percent class.

Our approach has three advantages :
(1) the integration of unsupervised neural learning and inductive machine learning in automated knowledge acquisition,
(2) a flexible, domain-dependent decision criterion for selecting significant attributes instead of a predetermined minimal decision criterion (as usual) in rule generation,
(3) the possibility for constructing rule conditions in various points of view.

The examples for learning may be incomplete and even inconsistent. Therefore the extracted rules should also be fault tolerant. A promising approach to this is to use a fuzzy set calculus [Enbutsu/91] [Mukaidono/92] [Ultsch/91b] [Weber/91] [Yi/92]. We will extend our ideas into that direction in the near future [Ultsch/91b].

Acknowledgement

We thank Mr. D. Korus, Ms. G. Guimaraes, and Mr. V. Weber for the helpful discussions. This reserach has been supported in part by the German Ministry of Research and Technology (BMFT) and by the Bennigsen-Foerde price of Nordrhein-Westphalia.

References

[Enbutsu/91]
Enbutsu, I., Baba, K., Hara, N. "Fuzzy Rule Extraction from a Multilayered Neural Network" in Proc. of IJCNN-91.
[Kohonen /89]
Kohonen, T. "Self-Organization and Associative Memory" 1989.
[Mukaidono/92]
Mukaidono, M., Yamaoka, M. "A Learning Method of Fuzzy Inference Data with neural Networks and Its Applications" in Proc. of 2nd. Int.Conf. on Fuzzy Logic and Neural Networks. 1992.
[Quinlan/83]
Quinlan, J.R. "Learning Efficient Classification Procedures and Their Application to Chess Endgames" in Machine Learning : An AI Approach, Vol. 1, Michalski, R. S., Carbonell, J.G., and Mitchell, T.M., eds., 1983.
[Ultsch/90]
Ultsch, A., Siemon, H.P., "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis" in Proc. of Int. Neural Networks Conf., 1990.
[Ultsch/91a]
Ultsch, A. "Konnektionistische Modelle und ihre Integration mit wissensbasierten Systemen" (in German) Research Report No. 396, Department of Computer Science of University of Dortmund, Germany, 1991.
[Ultsch/91b]
Ultsch, A., Höffgen, K.U., "Automatische Wissensakquisition für Fuzzy-Expertensysteme aus selbstorganisierenden neuronalen Netzen" (in German) Research Report No. 404, Department of Computer Science of University of Dortmund, Germany, 1991.
[Ultsch/92]
Ultsch, A. "Self-Organizing Neural Networks for Knowledge Acquisition" in Proc. of ECAI 1992.
[Weber/91]
Weber, R., Zimmermann, H.J. "Automatische Akquisition von unscharfem Expertenwissen" (in German) KI Magazin, No. 2, 1991.
[Yi/92]
Yi, H.J., Oh, K.W. "Neural Network-based Fuzzy Prodection Rule Generation and Its Application to an Approximate Reasoning Approach" in Proc. of 2nd. Int.Conf. on Fuzzy Logic and Neural Networks. 1992.