NEURAL NETWORK SYSTEM FOR KNOWLEDGE DISCOVERY IN DISTRIBUTED HETEROGENEOUS DATA

Timofeev A.V[1], Azaletskiy P.S.[1], Myshkov P.S[2], Kesheng Wang[3]
[1]Saint-Petersburg Institute for Informaion and Automation RAS, 14 line 39, St. Petersburg 199178 Russia; [2]Saint-Petersburg State University, 7-9 Univesrsitetskaya nab., St. Petersburg, 199034 Russia; [3]Norwegain Unversity of Science and Technology, N-7491, Trondheim, Norway. E-mail: Tav@iias.spb.su; kesheng.wang@ntnu.no.

Abstract: This paper proposed a distributed KDD system allowing remote usage of expert knowledge. Its implementation on the base of polynomial neural networks is described. The system is an universal KDD tool as it can build decision-making models in any subject field. Its implementation as a web-service will allow third-party software developers to create specialized applications, which focus on neural knowledge base usage.

Key words: knowledge discovery; polynomial neural networks; distributed KDD systems

1. INTRODUCTION

Modern computational systems and computer networks allow accumulation of huge amounts of data for solving problems in information analysis and optimal control. However, data in computer representation form contain information in a hidden way. To extract this information and knowledge, it is necessary to use special methods of data analysis and knowledge base synthesis.

Large amounts of data allow getting results that are more precise; but searching for solutions and knowledge synthesis is a very complicated task. As a result, a completely new class of systems serving as data analysis agents and knowledge engineers has emerged recently [2]. Such systems do

Please use the following format when citing this chapter:
Timofeev, A.V., Azaletskiy, P.S., Myshkov, P.S., Wang, Kensheng, 2006, in International Federation for Information Processing (IFIP), Volume 207, Knowledge Enterprise: Intelligent Strategies In Product Design, Manufacturing, and Management, eds. K. Wang, Kovacs G., Wozny M., Fang M., (Boston: Springer), pp. 144-151.

not simply analyze the data they contain, but also they are able to build decision-making models on the base of parallel data analysis. These systems are called Knowledge Discovery in Databases (KDD) systems [2].

The main task of KDD systems is analysis of data contained in databases with the purpose of discovering hidden, unobvious and unknown patterns and rules. The heterogeneous nature of modern distributed databases significantly complicates the task. Besides, each type of database requires different method of access to its data. Consequently, for each intellectual analysis system it is necessary to develop specific algorithms that take into account its features and characteristics. Modern databases are not only heterogeneous, but redundant as well. For a KDD synthesis, it is advisable to simplify the schema and the knowledge base structure as much as possible.

2. THE KDD TECHNOLOGY

Knowledge Discovery in Databases (KDD) is a process of search of meaningful knowledge in raw data [6]. The knowledge could be represented as a set of rules describing relations between data properties (decision trees), frequently encountered models (association rules), and classification results (neural networks) or data clusters (Kohonen maps), to name a few.

Regardless of knowledge representation model used, a KDD process consists of the following stages:

1. Data preparation. At this stage a data set is constructed, data consolidation is performed, and training and test sets are prepared.
2. Data preprocessing. Some tasks require input data be supplemented with certain information (e.g., expert information). Additionally, certain preprocessing algorithms, e.g. dimensionality reduction algorithms, are often required and executed at this step.
3. Data transformation and normalization. At this step the data is transformed into the form suitable for subsequent analysis.
4. Data mining. At the data mining step various knowledge discovery algorithms, such as neural networks, are executed.
5. Data postprocessing. Finally, different activities concerning interpretation of the results and their application to business are performed.

Knowledge Discovery in Databases does not determine what data analysis or processing algorithms should be used; instead, it defines the sequence of activities that should be performed in order to extract meaningful knowledge from source data. This approach is universal and does not depend on the application domain.

3. NEURAL NETWORK APPROACH FOR KDD

When using artificial neural networks (ANN) the choice of network architecture (the number of layers and the number of neurons in each layer) is of primary concern. The size and the structure of a network should correspond to the essence of the investigated problem (e.g., in terms of computational complexity). In the works [3-5] authors suggest three-tier polynomial and Diophantine neural networks, multi-tier "many-valued decision tree" neural network, as well as method for transformation of treelike neural networks into polynomial neural networks and for minimization of their complexity. Since at the beginning of the analysis the nature of the problem is often not well known, the choice of adequate ANN architecture becomes a rather complicated problem.

During the learning process a neural network uses input data to adjust its synaptic weights and to fine tune connections between neural elements. The resulting neural network expresses patterns present in the data and is able to make decisions on new data. The network serves as a functional equivalent for some data dependency model, i.e. a function of input variables, much like a one created using traditional modeling.

The principle advantage of neural network is that they are able to approximate any continuous function; therefore, there is no need for a researcher to assume any hypotheses regarding the model and sometimes even what variables are important.

A typical knowledge base construction system works that utilizes neural networks is shown on the Figure 1:

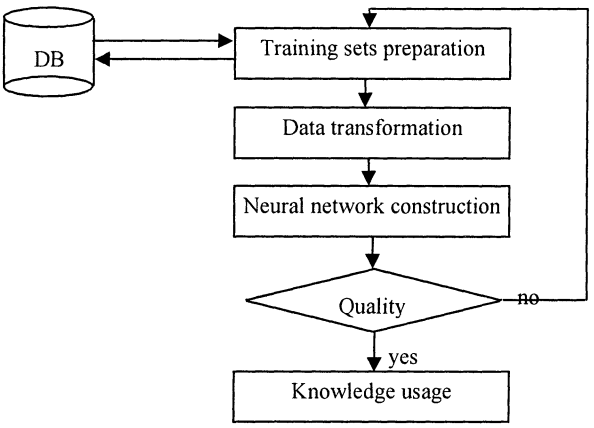

Figure 1. Sequence of operations for knowledge search with neural networks

3.1 Data sets preparation

At this stage qualified users or experts who possess knowledge in a certain application domain and want to automate the decision making process prepare a training set. This set $T = \langle Y, X \rangle$ is composed from known feature values $x_1, x_2, ..., x_n$ and decision function y values defined on an object set Ω:

$$\Omega = \{\omega_k : k = \overline{1, N}\}$$
$$Y = \{y(\omega_k) : k = \overline{1, N}\}$$
$$X = \{(x_1(\omega_k), ..., x_n(\omega_k)) : k = \overline{1, N}\}$$

The set T can be easily represented as a table:

Table 1. Training set table representation

ω	Y	x_1	...	x_n
ω_1	$y(\omega_1)$	$x_1(\omega_1)$	...	$x_n(\omega_1)$
...	...	...	...	...
ω_N	$y(\omega_m)$	$x_1(\omega_m)$	...	$x_n(\omega_m)$

Here $y(\omega_k)$ serves as an expected result of the problem solution on the input set $\{(x_1(\omega_k), x_2(\omega_k) ..., x_n(\omega_k)\}$. Training set is a database table with the expected values of decision function in the first column and input data feature values in the rest. The following constraints apply to this database:

- Validity: only qualified specialists take part in the creation.
- Completeness: all possible solutions are presented in the table.
- Consistency: there should not be two different values of decision function for the same input feature value.

These constraints allow to avoid noise influence, inconsistency and incompleteness in data representation.

3.2 Data transformation and normalization

If a KDD system uses neural networks, an expert (a user with strong knowledge in the application domain) has to fill in the training set. The system will then transform this input into numbers. Some systems use binary encoding of the set elements, i.e. {-1, +1}. If necessary, the training set could be sorted then.

3.3 Neural network construction and training

At this stage, a neural network is constructed and trained using various learning algorithms. As a result, a "neural knowledge base" is created.

3.4 Knowledge base quality control

After the neural network is trained, an expert tests its quality using training and test sets (these two sets should not intersect). The test set has the same structure as the training set. Based on the test results the expert makes decision on whether the constructed knowledge base is suitable for the given task or not. If not, the knowledge base construction process should be restarted from the step 1.

3.5 Knowledge base usage

To solve a task using the constructed knowledge base, a user prepares a table of feature values and sends it to the neural network input. The answer given represents the value of the decision function or the synthesized ANN.

3.6 Structure and organization of a distributed neural KDD system

The problem of development of a distributed system with remote knowledge base creation and usage capabilities is of high importance today. Such system often implements client-server architecture and uses Internet for information transmission.

A distributed neural KDD system operates in two main stages:

1. Neural knowledge base creation and control, implemented by an expert (the expert stage).
2. Knowledge base usage for the purpose of solving specific user tasks (the user stage).

A knowledge base system based on threshold polynomial neural networks has been developed [3-5]. It uses multiagent technology for processing and transmitting databases (DB) and knowledge bases (KB) through e-mail and HTTP protocols.

In the given model, experts producing a knowledge base and users working with it act as agents. The neural network type in this case can be described with arithmetic (Diophantine) polynomials. It is used for recognition of complicated (linearly non-separable) pattern types defined in a space of either binary or multi-valued feature-predicate. The scheme is illustrated on the Fig. 2.

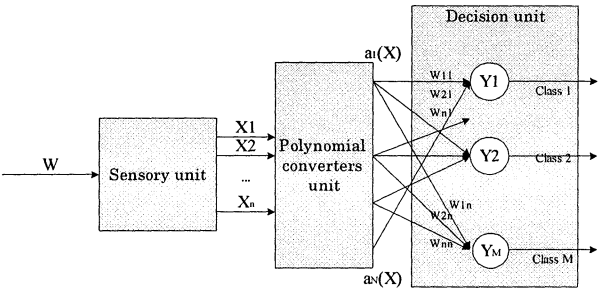

Figure 2. Structural scheme of a threshold polynomial neural network

At the input of a neural network there is a sensory unit consisting of threshold neuron-like elements, which encodes object features as binary codes $X = |x_1, x_2, ..., x_n|$.

The feature vector (X) is transmitted to a unit of polynomial converters (A-elements), which creates an m-dimensional vector of secondary (polynomial) features $Z = (a_1(X), a_2(X), ...a_N(X))$. These secondary features define a polynomial feature space $a_j(X), j = 1,2,...,n$, a so-called rectifying space. Explicit function form $a_j(X)$ is chosen in accordance with the given task and the training set, i.e. during the process of neural network construction and self-organization [3-5].

The output tier contains solution threshold neuron-like elements:

$$Y_i = sign(\sum_{j=1}^{N} w_j a_j(x)), \text{ where } i \in [1..M]$$

The recurrent learning algorithm used in threshold polynomial neural network is a supervised learning algorithm, which offers a number of advantages over the frequently used back-propagation of error (BPE) [1], including the following:

1. It is not necessary to determine network structure in advance, since the algorithm adjusts itself during learning process.
2. This is a single-pass algorithm, i.e. the third layer (decision layer) neurons' weights are adjusted in the first pass through the training set.
3. The algorithm guarantees error-free classification of elements in the training set.
4. The algorithm constructs a neural network with a high degree of extrapolation to data beyond the training set.

5. The synthesized neural network allows creation of a neural knowledge base based on the source database.

The structural scheme of the distributed KDD system is shown on the Figure 3:

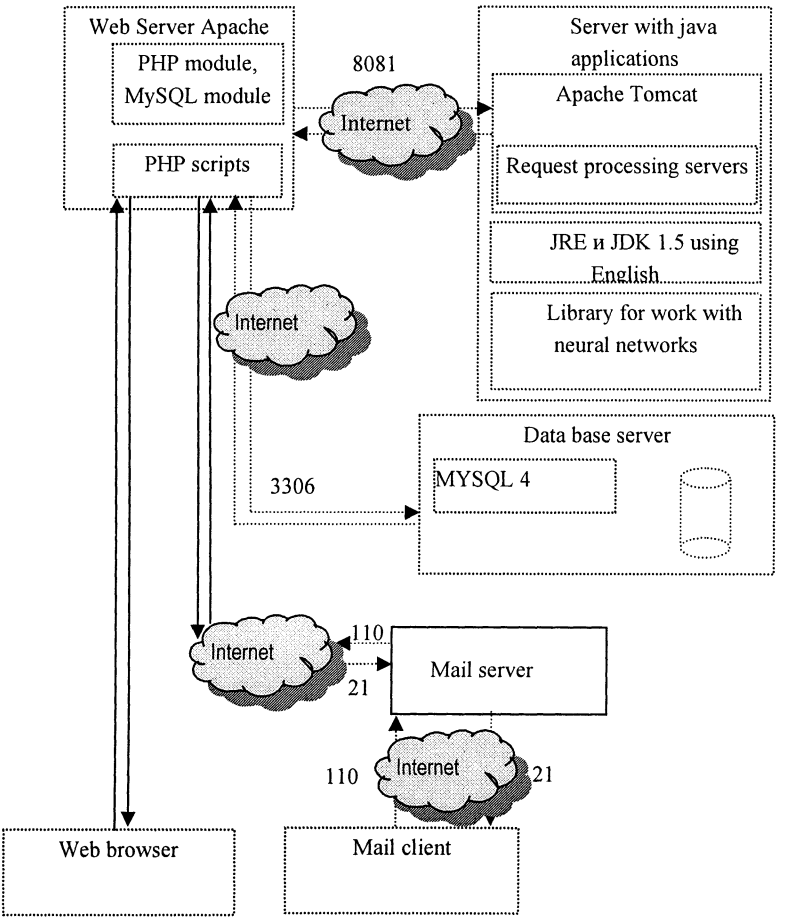

Figure 3. Structural scheme of the distributed neural KDD system.

At the first stage, an expert creates training and test sets in the form of database table. For this task, an expert query is constructed. The second stage concerns work with the knowledge base, at which a user query is constructed. As a reply, the system fills in the "Result" value corresponding to the certain feature cortege, which serves as an output generated by the system in response to the user query.

4. CONCLUSION

The distributed KDD system proposed in the article allows remote usage of experience of experts in a certain field, and its implementation as a neural knowledge base. The system is a universal KDD tool, since it makes it possible to build decision-making models in any subject field. Its improvement to a web-service will allow third-party software developers to create specialized applications oriented on neural knowledge base usage. The system can be applied in research as well: as a tool for in-depth study of different effects in ecology, economics and other fields. It serves as means to integrate problem solving experience of geographically distributed users. Apparently, the system could be useful in telemedicine area (for example, to help get the most probable diagnosis of complicated and rare illnesses), in the area of financial flows analysis and many others.

5. REFERENCES

1. Lugger, G. F. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 2003, p. 864.
2. Bagresan, A. A., Kuprianov, M. S., Stepanenko, V. V., Holod, I. I. *Methods and Models of Data Analyzing: OLAP and Data Mining.*
3. Simon Haykin, *Neural Networks a Comprehensive Foundation*, 2006, p. 1104.
4. Timofeev, A. V., Methods of Creation of Diophantine Neural Networks with Minimal Complexity – *RAS Report*, 1995, Vol. 345 No.1, pp. 32-35.
5. Timofeev, A. V., Parallelism and Self-Organization in Polynomial Neural Networks for Image Recognition – *Proceedings of the 7th International Conference on Pattern Recognition and Image Analysis: New Information Technologies* (18–23 October, 2004, St. Petersburg), pp. 97-100, 2004.