Complete Round-Robin method for data mining

Boris Kovalerchuk, Clayton Todd, Dan Henderson
Dept. of Computer Science, Central Washington University, Ellensburg, WA, 98926,
borisk@tahoma.cwu.edu, hendersond@cwu.edu, toddc@cwu.edu

Abstract.
The reliability of data mining applications heavily depends on testing of discovered patterns. The common testing approach is based on the Round-Robin method for available data. Usually training data are randomly chosen from the available data and the rest of the data are used for testing. Each time, the parameters of the system are identified using training data and then correctness of these parameters is tested on testing data. There are $2^n$ pairs of training and testing data sets for $n$ data objects. Actual Round-Robin implementations heuristically restrict this number of combinations. It decreases the amount of computations, but these smaller tests may be not valid to represent all $2^n$ possible tests.

We developed a method to speed up a complete Round-Robin computational procedure using the concept of monotonicity and multithreaded programming for Windows NT. This method is applicable for both attribute-based learning and relational data mining methods. We implemented this method and show its effectiveness for Neural Networks based on backpropagation.

1. Introduction
Data Mining methods are used for tasks ranging from optical character recognition to medical diagnosis. The reliability of these methods depends on how well the discovered patterns are tested before they are used for their intended purpose. One common approach used to test Neural Networks is to divide the data set into two parts. Approximately 30% of the data are chosen at random and this subset becomes the testing data and is used to validate the patterns discovered by the data mining method after processing the remaining data. While this does cut down the number of calculations that the method needs to perform while training, the testing subset may not be representative of the data set as a whole. A more comprehensive approach is to train the network and then test it on all possible subsets of the original data set. The problem with this approach is that there are $2^n$ possible subsets, where $n$ is the number of objects in the data set. The purpose of this study is to design and implement a method that will speed up these computations.

2. Concepts and Definitions
Let $M$ be a Data Mining Method, and let $D$ be a set of $n$ objects, represented by $m$ attributes. $D=\{d_i\}, i=1, n, d_i=d_{i1}, d_{i2}, \ldots, d_{im}$. Method $M$ is applied to data $D$ for knowledge discovery.

We assume that data are grouped, for example in time series first 250 data objects (days) belong to 1980, next 250 objects (days) belong to 1981 and so on.
To simplify consideration, we assume that we have ten of these groups (years). There are $2^{10}$ subsets of them. Any of them can be used as training data and their complements can be used for testing. We also considered completely independent data for testing.

**Hypothesis of monotonicity (HM).**

If $D_1 \supseteq D_2$ then $P(M,D_1) \geq P(M,D_2)$ \hspace{1cm} (1)

This hypothesis means that if data set $D_1$ covers data set $D_2$ then performance of method $M$ on $D_1$ should be better than or equal to performance of $M$ on $D_2$. Basically, this hypothesis assumes that extra data bring useful information for knowledge discovery. Obviously, the hypothesis is not always true. We tested this hypothesis in data sets from some applications. It was confirmed reasonably. We coded all $2^n$ data subsets as binary vectors $v_i = (v_{i1}, v_{i2}, \ldots, v_{i10})$ with 10 components from 00000 00000 to 11111 11111. In these binary terms, the hypothesis of monotonicity can be rewritten:

If $v_i \geq v_j$ then $P(M,D_1) \geq P(M,D_2)$, \hspace{1cm} (2)

where $v_i \geq v_j \iff v_{ik} \geq v_{jk}$ for all $k=1,m$.

Let us also introduce quality indicator $Q(P,D_1) = 1 \iff P(M,D_1) > Q_0$, where $Q_0$ is an acceptable performance limit. This assumption allows us to transform (2) to (3).

If $v_i \geq v_j$ then $Q(M,D_1) \geq Q(M,D_2)$ \hspace{1cm} (3)

In this way we obtain $Q(\ , \ )$ as a monotone Boolean function. The theory of monotone Boolean functions [Kovalerchuk et al., 1996] is a well-developed theory with mechanisms to speed up computations. We exploit this theory to speed up Round-Robin method.

3. Result

According to HM hypothesis if method $M$ does not perform well on data $D_1$, it also cannot perform well on $D_2$. Therefore, we do not need to test method $M$ on $D_2$. Our experiments show that, eliminating these redundant computations, it is possible to run method $M$ 250 times instead of all 1024 times. We also exploit Hansel chains and lemma [Kovalerchuk, et al., 1996] from theory of Monotone Boolean functions to optimize the sequence of testing method $M$ on different data subsets. Using this theory we developed a multithreaded application in C++. The program is designed in such way that it can run in parallel on several processors to further speed up computations. A screen shot of the implementation for backpropagation Neural Network is presented in figure 1.

**Figure 1. Main screen**

**Reference**