Parallel GA-based Wrapper Feature Selection for Spectroscopic Data Mining

Nordine Melab
Laboratoire d’Informatique du Littoral
Université du Littoral - Côte d’Opale
50, rue Ferdinand Buisson - BP 719
62228 - Calais cedex - France
E-mail: melab@lil.univ-littoral.fr

Sébastien Cahon and El-Ghazali Talbi
Laboratoire d’Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille1
59655 - Villeneuve d’Ascq cedex - France
E-mail: {cahon,talbi}@lifl.fr

Ludovic Duponchel
Laboratoire de spectroscopie infrarouge et raman de Lille
Université des Sciences et Technologies de Lille1
59655 - Villeneuve d’Ascq cedex - France
E-mail: ludovic.duponchel@univ-lille1.fr

Abstract

Mining predictive models in dense databases is CPU time consuming and I/O intensive. In this paper, we propose a taxonomy of existing techniques allowing to achieve high performance. We propose a hybrid approach allowing to exploit four of them: feature selection, GA-based exploration space reduction, parallelism and concurrency. The approach is experimented on a near-infrared (NIR) spectroscopic application. It consists of predicting the concentration of a given component in a given product from its absorbances to radiations. Statistical methods, like PLS, are well-suited and efficient for such data mining task. The experimental results show that preceding those methods with a feature selection allows to withdraw a significant number of irrelevant features and at the same time to enhance significantly the accuracy of the discovered predictive model. It is also shown that for the considered task the GA-based approach allows to build more accurate models than neural networks. Moreover, the parallel multi-threaded implementation of the approach allows a linear speed-up.

Keywords: Feature Selection, Genetic Algorithms, Parallelism, Concurrency, NIR spectroscopy.

1. Introduction

Data mining is a process of discovering in databases novel, accurate, useful and comprehensive predictive models. The databases are often dense meaning their schema is composed of a large number of attributes. Considering all the features to build models poses two major problems: first, data exploration is CPU time consuming. Indeed, theoretical and practical studies [20] have shown that it is in general exponential in the number of features. Second, some features are irrelevant and/or redundant. They do not help for the comprehension of the discovered models. In the literature, several definitions of relevance have been proposed. For our problem, a feature is considered relevant if its elimination causes for the discovered predictive model a significant loss of accuracy. A feature is said redundant if it is correlated with other features, and so it does not bring any complementary information for the predictive model. Those two questions arising when dealing with dense databases has led researchers to reduce the dimensionality of the problem by sampling the raw data. Feature selection [11] is one approach allowing that reduction. The task is to select $M$ relevant and/or no redundant features from $N$ originally given features where $M \leq N$ without decreasing the accuracy of the model. Optimal feature selection is proved to be NP-hard [16].

Several techniques attempting to solve the problem have been proposed in the literature [16, 18, 5, 11]. These approaches may be classified in two categories [10]: filter approaches and wrapper approaches. Filter techniques consider two separated steps: the first one performs the feature selection; the second step builds the model from the selected attributes. The selection is not based on the accuracy of the built model but on some criteria such as the correlation between the features. Conversely, in the wrapper methods the attribute selection and the model building
are mixed. The attribute selection is performed in several steps. At each step, a model is built and its accuracy is evaluated. The accuracy informs about the quality of the performed selection: more accurate is the model more interesting is the selection. The selection is enhanced from step to step until no accuracy improvement is possible. In each category, different techniques have been proposed. In this work, we focus on the genetic algorithms (GAs). GAs have been revealed as a powerful tool for attribute selection [19]. A survey on genetic feature selection in mining issues is proposed in [12].

On the other hand, in order to reduce the complexity of the model-building process the selection has to be done in parallel. Moreover, the process is I/O intensive. The access data during the model building has thus to be optimized. Concurrency is known to be a good way to achieve high-performance in I/O intensive applications.

In the present paper, we propose a taxonomy of high-performance data mining (e.g. models building) techniques. We then particularly focus on a parallel multi-threaded wrapper GA-based feature selection technique, namely $W = GAFS$. In order to evaluate the pertinence of our work, we consider a NIR spectroscopic application. NIR spectroscopy is an analytical method allowing economical, efficient and reliable quality control. It is utilized in industry to determine the composition of products, to predict their concentrations, etc. The focus here is predicting the concentration of a given component in a given product. In particular, we deal with predicting the concentration of sugar in beet. The predictive model is built from a set of data samples. Each sample contains the values of the absorbances of the product to NIR radiations. Each absorbance value corresponds to a wavelength belonging to the domain $830nm \rightarrow 2500nm$. For each sample, a concentration of the studied component is measured by chemical analysis. The feature selection problem consists here of identifying the relevant wavelengths that allow to predict the concentration of a given component in a given product. $W = GAFS$ is coupled with the Partial Least Square (PLS) [17] to solve the problem. In order to improve its efficiency a parallel multi-threaded implementation is proposed.

The rest of the paper is organized as follows: Section 2 highlights the different ways to achieve high-performance data mining. Section 3 describes the studied spectroscopic problem and the proposed approach to solve it. Section 4 presents the parallel multi-threaded implementation of the approach. In Section 5, an experimentation of the spectroscopic data mining application is presented, and the results are discussed. Finally, Section 6 concludes the paper and draws some perspectives.

2. High-Performance Data Mining

Data mining can be viewed as a process allowing to explore a space of potential solutions (knowledge) in order to discover the most interesting of them. As Figure 1 illustrates it, this space is built from the database schema meaning its different features.

![Figure 1. Different possible sources for high-performance data mining](image)

During exploration, the miner evaluates the quality (interestingness) of the knowledge by accessing the database on the hard disk. For dense databases (thousands of features), such as spectroscopic ones, achieving performant exploration becomes a hard problem.

According to Figure 1, three parameters have to be considered if one has to deal with the performance problem: the database, the exploration space and the data access. Consequently, five classes of approaches, presented in Figure 2, can be distinguished: data-oriented approaches, algorithmic-oriented ones, parallelism, data access optimization and active data mining.

All these approaches aim at speeding up the mining process, and at the same time maintaining as higher as possible the interestingness of the discovered knowledge. They can be summarized as follows:

- Data-oriented approaches attempt to reduce the amount of data to be accessed. To do that three techniques are often utilized: feature selection [11], data sampling [21], domain discretisation [9]. First, feature selection aims at limiting the number of features to appear in the knowledge. The selection can be done randomly or on the basis of some criteria (e.g. correlation between features) allowing to choose the useful (relevant and/or redundant) features. Second, data sampling techniques reduce the number of data instances
(database entries) to be considered to a small set (training set). That set can be determined randomly or with some criteria (e.g. the date of the instances i.e. cash tickets if the database is a commercial one). Third, domain discretisation means that the domain defining the possible values of some features is projected on a smaller one. The discretisation method depends strongly on the data mining application. It is particularly hard for features with continuous values.

- Algorithmic-oriented approaches focus on the space of candidate knowledge. Two techniques can be distinguished here: code optimization-based ones [1] and evolutionary algorithms [6]. The objective of the first category is to optimize the space exploration time. It is based on efficient data structures. For example, in [8] hash trees are utilized to store the potential association rules. Evolutionary algorithms consider another approach that consists of limiting the amount of potential knowledge to a given sub-space (population) initially randomly selected. As in natural evolution the selected potential knowledge are replaced from generation to generation by other ones that are more interesting.

- Parallelism can be either control-guided or data-guided [6]. Control parallelism consists of partitioning the exploration space and allocating the resulting partitions to the different nodes composing the parallel machine. The exploration is thus performed in parallel. Data parallelism can be exploited when the database is divided into several partitions each of them placed on one node of the parallel machine. As the interestingness of the potential knowledge is often evaluated on the whole database these ones have to visit all the nodes. Consequently, the measurement of the knowledge quality is data parallel.

- Data access optimization allows to speed up the access to the data on the hard disk. To achieve that three ways are possible: out-of-core computing mining [13], utilization of DBMS facilities [3] and concurrency [13]. First, out-of-core mining is necessary when the space of potential knowledge is very large to make it impossible to fit into the memory of the machine. A part of it has thus to stay on the hard disk, and then to be loaded exactly at the moment their use is required. An intelligent strategy consists of identifying this part and the moment it has to be loaded at run-time. Second, DBMS facilities allow to exploit the long experience purchased in the DBMS optimization field. Stored procedures and user-defined functions (UDF) are two important ways to minimize the overhead induced by the exchange of data instances between a DBMS and a miner. Third, concurrency is often neglected by the data mining community although it is very important. Indeed, it can be efficiently used to recover the I/O operations by computation. When data is accessed by threads, other threads can explore the potential knowledge and evaluate their interestingness.

- Active data mining [2] is important when the database to be mined is active meaning that new data instances arrive continually in that database. The problem that can arise is that the new data could make the knowledge previously discovered invalid. A simple solution consists of periodically performing the data mining process on the whole database (old and new data). However, this solution becomes sooner or later very costly with the increase in size of the database. The best solution, as it is proposed in [3], is that allowing to perform the data mining process only on the
new data. The discovered knowledge have to be compared to the old knowledge (stored in a knowledge base with their interestingness values). Those having significantly changed must be updated.

Finally, to achieve high-performance data mining we believe that, as in [14], a hybridization of different techniques is required. In this paper, we consider together the approaches in bold in Figure 2 i.e. attribute selection, evolutionary algorithms, control parallelism and concurrency.

3. GA-based Feature Selection

3.1. The studied problem

We recall that in this paper, we deal with the problem of building a model allowing to predict the concentration of a given component in a given product. To do that, NIR radiations ($N$ wavelengths) are sent through a sample of the product (see Figure 3). Sensors are then utilized to measure the absorbances by the product of those radiations. A spectrum of $N$ absorbances is thus obtained. In addition, the concentration of the component in the product is measured by chemical analysis. The experience is renewed a certain number $M$ of times. Therefore, a set of $M$ samples is obtained. Each sample contains a spectrum of $N$ absorbances and its corresponding real concentration. The set of samples constitute the data of the studied problem. In order to check the accuracy of the model, once built, the data set is divided into two sub-sets: a calibration sub-set and a validation sub-set. They allow respectively the building of the model and its validation.

![Figure 3. Measurement of the radiations absorption in the near-infrared domain](image)

According to the Beer Lambert law, the absorbance for a given wavelength is proportional to the sum of the concentrations associated with the components that absorb it. Therefore, the variation of the concentration of a given component acts on the absorbance associated with the wavelengths that component can absorb. In other words, the concentration of the component is a function of the absorbances corresponding to the wavelengths it can absorb. Predicting the concentration of a given component in a given product consists of estimating that function (model). As the problem is linear (due to the Beer Lambert law), it is classically solved by statistical methods. The well-suited for NIR spectroscopy is the PLS regression method [17]. Moreover, once the predictive model is built from the calibration data sub-set its accuracy is evaluated on the validation data sub-set by using the following metric, called Root Mean-Square (RMSEP):

$$RESEP = \sqrt{\sum_{i=1}^{n} (c_i - \hat{c}_i)^2} / n$$

where $n$ is the number of validation data samples. For each sample $i$, $c_i$ and $\hat{c}_i$ designate respectively its measured (real) concentration and its predicted concentration (computed with the built model).

This paper deals with a particular instance of the problem: predicting the concentration of sugar in beet. The data set is provided by the Laboratoire de spectrochimie infrarouge et raman de Lille. The calibration and validation sub-sets contain respectively 957 and 1024 samples. Each sample contains 1020 absorbances (corresponding to 1020 wavelengths) and the associated concentration measure determined by chemical measurement.

3.2. Genetic feature selection

We believe that before applying PLS, a feature selection has to be performed. Before our approach is presented let us examine why this selection is needed for the application this paper deals with. We measured on the absorbances of the calibration samples the two parameters: redundancy and relevance. Figure 4 shows the average correlation of the absorbance corresponding to each wavelength with the absorbances associated with all other wavelengths. The average values are between 0 and 1. Features (absorbances) with a high average correlation are redundant. Those around 400nm and 800nm are particularly redundant.

On the other hand, Figure 5 illustrates the correlation of the absorbances with the concentration. The values of the correlation are between $-1$ and 1. The values near to the extremities define a perfect correlation i.e. a strongly relevant features. Conversely, features with a correlation near to 0 are irrelevant. One can remark that the features that correspond to the wavelengths between 0 and over 350 are particularly irrelevant.

Given the above measurements, before applying the PLS procedure one has really to perform a feature selection. The objective is to select the absorbances that are less correlated between them and more correlated with the concentration.

The feature selection problem can be formulated as follows: a data sample can be viewed as an assignment of absorbance values to a set of wavelengths ($w_1, w_2, ..., w_n$)
with a concentration $c$. If $f$ represents the model to be built, $c = f(w_1, w_2, \ldots, w_n)$. In presence of irrelevant and/or redundant wavelengths, feature selection consists of selecting $M$ relevant wavelengths from $N$ given wavelengths where $M \leq N$. As it is shown in Figure 6, our method is a wrapper GA. The individuals of the population are binary strings with the length equal to the total number of considered wavelengths. Each individual is a string of bits $w = w_i$, $i = 1, 2, \ldots, N$ in which $w_i = 1$ means that $w_i$ is relevant, thus selected, and $w_i = 0$ means that $w_i$ is irrelevant, thus not selected. At each generation, a $PLS$ is performed for each individual i.e. selection according to its absorbances (wavelengths). The fitness of the individual is the prediction error $RMSEP$ of the model being built from its corresponding selection. The genetic operators are the classical ones meaning the mutation and the crossover. In this work, the crossover operator has two schemes: uniform and one-point.

4. Parallelization and Concurrency

During the evaluation step of $W - GAFS$, for each individual (feature selection) a model is built and its $RMSEP$ is computed. Moreover, some preliminary experiments have shown that $PLS$ is costly in terms of CPU computation time and I/O operations. Indeed, the execution of $PLS$ on an IBM RS/6000 375MHz takes about 10 seconds. Therefore, the evaluation of the individuals must be parallel. In addition, in order to overlap the I/O operations by the computation concurrency is required.

The definition of the parallel model depends strongly on the partitioning and the placement of the population and the database. The population can be either centralized on one processor or distributed among several processors. On the other hand, the database can be centralized, replicated or distributed. Table 1 summarizes the different possible combinations and the advantages (+) and drawbacks (-) of each configuration. Computing consists of applying the $PLS$ method according to the individuals. I/O operations are mainly performed during the application of $PLS$.

Therefore, defining a parallel model requires to take into account the advantages and drawbacks of each configuration. It consists of finding the best compromise between the different configurations. According to Table 1, the population must be distributed. Regarding the database, as it is not large in our case its distribution would not be beneficial. Its replication allows I/O parallelism but a loss of disk space. On the other hand, its centralizing causes trottling situations, thus performance decreasing. The best way to achieve I/O efficiency would be the hybridization of the two configurations. That consists of logically partitioning the parallel (or distributed) machine into clusters. The database is replicated on each cluster, and is centralized for the processors belonging to the same cluster. The idea is developed in the present work.

As Figure 7 illustrates it, our parallel model is the $SPMD$ one. The farmer creates the workers on $N$ nodes (one worker per node) and sends them the database. Each node (worker) represents a cluster. Then the farmer divides the population into sub-populations, and each of them is affected to one worker. Each worker evaluates the individuals of its sub-population in parallel on the processors of its cluster. Those ones perform the $PLS$ procedure by accessing the database of their cluster. Once all the individuals are evaluated, the worker returns back the results.
to the farmer which performs the replacement operation on the whole new population. The cycle is repeated during a certain number of generations.

5. Experimentation and Discussion

From the GA side, the population is composed of 200 individuals, the probabilities of applying the crossover and mutation are fixed as equal to respectively 80% and 10%. Usually, the mutation probability is fixed to 0.1%, but it is not sufficient in our case as the chromosomes are long (1020 genes). The GA performs 2000 generations.

The experimentation material platform is an IBM SP2 composed of 4 nodes (clusters) of 16 processors. The programming environment is the Parallel Virtual Machine (PV M) [7] together with the Posix threads library (pthreads) [15]. PV M and pthreads allow respectively the parallel and multi-threaded coding of \( W - GAFS \).

In order to evaluate the impact of the feature selection on the accuracy of the built models, we experimented the parallel multi-threaded \( W - GAFS \) on the calibration set of samples (see 3.1). The obtained results are illustrated in Figure 8.

One may remark that the one-point and uniform crossover operators allow to build a model with a prediction error of respectively 0.14 and 0.124. In other words, the model built with an uniform crossover-based GA is over 11.4% more accurate than that built with the one-point crossover-based GA. This is because the chromosomes are long. In addition, other results demonstrate that the application of \( PLS \) without a genetic attribute selection allows to discover models with a prediction error of 0.17. Therefore, our approach enhances the accuracy of the built model by a magnitude of 27.6%. Moreover, the \( PLS \) algorithm.

---

**Figure 6. The wrapper GA-based attribute selection method**

**Table 1. Different possible parallel configurations**

<table>
<thead>
<tr>
<th>Database \ Population</th>
<th>Centralized</th>
<th>Distributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>No parallelism</td>
<td>+ Parallel computing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Throttling situation with I/O</td>
</tr>
<tr>
<td>Replicated</td>
<td>+ Parallel I/O</td>
<td>+ Parallel computing and I/O</td>
</tr>
<tr>
<td></td>
<td>- Loss of disk space</td>
<td>- Loss of disk space</td>
</tr>
<tr>
<td>Distributed</td>
<td>+ Parallel I/O</td>
<td>+ Parallel computing and I/O</td>
</tr>
</tbody>
</table>
has been coupled with neural networks in [4]. The experimental results show that the best possible prediction error is over 0.16. The model-building technique is 22.5% less accurate than our proposed approach.

On the other hand, a key topic that has to be considered in GA-based feature selection is the similarity of the sets of features selected at different runs. Ten runs have been performed and the visualization of the selected sets of features allows to observe their similarity. For example, Figure 9 and Figure 10 illustrate the sets of wavelengths selected at respectively the second and third runs. The numbers of relevant wavelengths are respectively 272 and 301. The average percentage of withdrawn irrelevant wavelengths is about 60%.

Finally, the parallel multi-threaded implementation of \( W - GAFS \) allows a linear speed-up. This is because no communication is performed between the workers at run time, and the I/O operations are recovered by the computation.

### 6. Conclusion and Future Work

Mining dense databases is known to be CPU time consuming and I/O intensive. In this paper, we proposed a taxonomy of high-performance techniques. We then focused on four of them: attribute selection, evolutionary computation, parallelism and concurrency. We proposed a parallel multi-threaded hybrid approach \( W - GAFS \) allowing to use the four ways to achieve high performance. The approach has been experimented in the \( NIR \) spectroscopy field. The application consists of predicting the concentration of sugar in beet from its absorption characteristics of \( NIR \) radiations. Due to the linearity of the problem, statistical methods, especially \( PLS \), are particularly efficient and useful. The obtained experimental results show that a feature selection must be prior to the building-model technique (here \( PLS \)). Indeed, it allows to withdraw 60% of irrelevant wavelengths, and at the same time an accuracy gain of 27.6%. In addition, the parallel multi-threaded implementation allows a linear speed-up.

Finally, optimal feature selection can be viewed as a multi-objective optimization problem that consists of selecting a minimum subset of features following three major objectives: (1) minimizing the average correlation of each feature with others; (2) maximizing the average correlation of the selected attributes with the knowledge to be predicted; (3) maximizing the size of selections, otherwise the solution will converge to a selection with only one feature. In the future, we will propose a GA-based multi-objective filter approach to deal with that multi-objective problem. That will allow us to compare the performance of the two approaches in terms of computation efficiency and quality of discovered models. Moreover, we will also develop a parallel implementation of the two approaches based on the island model in order to enhance the quality of the performed selection.
Figure 9. Selection obtained at the second run

Figure 10. Selection obtained at the third run

References


