Classification with Feature Selection via Mathematical Programming

STANISLAV BUSYGIN and PANOS M. PARDALOS
Department of Industrial and Systems Engineering
University of Florida
Gainesville, FL 32611, USA
{busygin,pardalos}@ufl.edu

Abstract: - Let a set of training and test samples be given, and the samples from the training set be partitioned into a number of classes, while classification of the test samples is unknown. The classification problem consists in determining classes of the test samples utilizing the information provided by the training set. Usually, not all features of the data set are informative for discovering the classification, and a subset of features relevant to it should be found. This task is called the feature selection. We handle it from the viewpoint of mathematical programming in the following way. We consider several unsupervised clustering principles and use them as constraints, while representing the desirable properties of feature selection as the objective function. In particular, we consider k-means local optimality constraints, pairwise threshold constraints, and biclustering consistency constraints. The involved objectives are used either to maximize separation of classes or to minimize the information loss.
The developed optimization-based approach has shown good performance on well-known DNA microarray data sets.

Key-Words: Data mining, Feature selection, Clustering, Classification, Supervised learning, k-means, Optimization, Biomedicine, Microarrays.

1 Introduction

Let a data set of n samples and m features be given as a rectangular matrix $A = (a_{ij})_{m\times n}$, where the value a_{ij} is the expression of i-th feature in j-th sample. We consider classification of the samples into classes

$$\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_r,\ \mathcal{S}_k \subseteq \{1 \ldots n\},\ k = 1 \ldots r,$$
$$\mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_r = \{1 \ldots n\},$$
$$\mathcal{S}_k \cap \mathcal{S}_\ell = \oslash,\ k, \ell = 1 \ldots r,\ k \neq \ell.$$

The set of samples is divided into the *training* and *test* sets. For the samples from the training set the classification is known, while for the samples from the test set it has to be performed utilizing the information provided by the training set classification. Generally, the classification should be done so that samples from the same class share certain common properties characterizing the classes.

This is one of the central problems of data mining theory and applications, and in practice it is frequently complicated by the presence of *outliers* (i.e., samples which do not possess characteristics of the majority of samples from their class) in the training set. Furthermore, usually not all features of the data are informative for discovering the classification, and a subset of features determining it should be found. This task is called the *feature selection*.

In this paper, we develop a mathematical programming approach to these major data mining problems. We make use of principles of unsupervised learning (clustering) and involve them in constraints of an optimization problem for feature selection. The objective function is formed to represent the goal of either maximization of class separation or minimization of the information loss. As feature selection is accomplished, the classification of test set samples is performed on the basis of the same unsupervised clustering principles that were used for feature selection constraints.

The paper is organized as follows. In the next section we consider a number of conditions that may be used to show that classes of samples are well-separated. They are normally used as stopping criteria of unsupervised clustering. In Section 3, on the basis of these criteria, we formulate our optimization-based algorithms for feature selection and classification. In Section 4 we present our computational experiment results on two well-known microarray data sets. Finally, in Section 5 we conclude the paper with general remarks and directions for further research.

2 Unsupervised Clustering Principles

Let us describe the formal setup for performing the feature selection. Let each sample be already assigned somehow to one of the classes $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_r$. Introduce a 0–1 matrix $S = (s_{jk})_{n\times r}$ such that $s_{jk} = 1$ if $j \in \mathcal{S}_k$, and $s_{jk} = 0$ otherwise. The sample class *centroids* can be computed as the matrix $C = (c_{ik})_{m\times r}$:

$$C = AS(S^T S)^{-1}, \qquad (1)$$

whose k-th column represents the centroid of the class $\mathcal{S}_k$. Each value c_{ik} in the matrix C gives us the average expression of the i-th feature in the sample class $\mathcal{S}_k$.

We also introduce a vector of variables $x = (x_i)_{i=1\ldots m}$ bounded between 0 and 1 representing chosen feature weights. If $x_i = 0$, then the i-th feature is disregarded during the test set classification.

The three types of constraints discussed next were used in the present research. We should point out that any of them is not universal and is applicable only to data of particular properties. However, there is no limitation as to what unsupervised clustering principles to use in the developed mathematical programming framework, and other suitable constraints may be involved if it is required by properties of the data (e.g., noisiness or incompleteness).

2.1 k-means local optimality

The given partition of the samples into the classes is (locally) optimal with respect to k-means if

$$\sum_{i=1}^{m} \left(a_{ij} - c_{i\hat{k}}\right)^2 x_i \le \sum_{i=1}^{m} \left(a_{ij} - c_{ik}\right)^2 x_i \qquad (2)$$

whenever $j \in \mathcal{S}_{\hat{k}}$, $\hat{k}, k = 1 \ldots r$, $\hat{k} \neq k$. Indeed, the $n \cdot (r-1)$ inequalities (2) imply that each sample is at least as close to the centroid of its class as to the centroid of any other class.

2.2 Pairwise threshold constraints

We can tighten the k-means local optimality constraints imposing the requirement that the distance between any two samples that belong to the same class is always not greater than any distance between two samples from different classes. This can be achieved by $\frac{n\cdot(n-1)}{2}$ inequalities of the form

$$\sum_{i=1}^{m} \left(a_{ij_1} - a_{ij_2}\right)^2 x_i \le D_{int} \qquad (3)$$

if samples j_1 and j_2 are from the same class, or

$$\sum_{i=1}^{m} \left(a_{ij_1} - a_{ij_2}\right)^2 x_i \ge D_{ext} \qquad (4)$$

if samples j_1 and j_2 are from different classes, with one additional inequality

$$D_{int} \le D_{ext}. \qquad (5)$$

We will call the inequalities (3)-(5) the *pairwise threshold constraints*. It is easy to see that the pairwise threshold constraints imply the k-means local optimality, but not vice versa.

2.3 Consistent biclustering

The last principle we use for feature selection constraints is based on simultaneous clustering of samples and features of the data set. Suppose there exists a partition of features into r classes

$$\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_r,\ \mathcal{F}_k \subseteq \{1 \ldots m\},\ k = 1 \ldots r,$$

$$\mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_r = \{1 \ldots m\},$$

$$\mathcal{F}_k \cap \mathcal{F}_\ell = \oslash,\ k, \ell = 1 \ldots r,\ k \neq \ell$$

such that features of class $\mathcal{F}_k$ are highly expressed in the samples of class $\mathcal{S}_k$. We will call the set of class pairs

$$\mathcal{B} = ((\mathcal{S}_1, \mathcal{F}_1), (\mathcal{S}_2, \mathcal{F}_2), \ldots, (\mathcal{S}_r, \mathcal{F}_r)) \qquad (6)$$

a *biclustering* of the data set. Similarly to the matrices S and C, we introduce the 0–1 matrix $F = (f_{ik})_{m \times r}$ such that $f_{ik} = 1$ if $i \in \mathcal{F}_k$ and $f_{ik} = 0$ otherwise, and the matrix of feature class centroids $D = (d_{jk})_{n \times r}$:

$$D = A^T F (F^T F)^{-1}, \qquad (7)$$

whose k-th column represents the centroid of the class $\mathcal{F}_k$. Now the value d_{jk} gives us the average feature expression in the sample j among features of the class $\mathcal{F}_k$. The condition of up-regulation of the features of a class $\mathcal{F}_k$ in the samples of the class $\mathcal{S}_k$ implies

$$i \in \mathcal{F}_{\hat{k}} \ \Rightarrow\ \forall k = 1 \ldots r,\ k \neq \hat{k}:\ c_{i\hat{k}} \geq c_{ik}, \qquad (8)$$

and, symmetrically,

$$j \in \mathcal{S}_{\hat{k}} \ \Rightarrow\ \forall k = 1 \ldots r,\ k \neq \hat{k}:\ d_{j\hat{k}} \geq d_{jk}. \qquad (9)$$

If the biclustering $\mathcal{B}$ satisfies both (8) and (9), we will call it *consistent*.

For the purpose of feature selection, when the classes of samples $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_r$ are already given, we construct the classes of features $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_r$ according to (8). Then, to obtain a consistent biclustering, we remove some features from the data set in order to satisfy (9). Considering now the variables x_i to be 0–1 (i.e., fractional feature weights are impossible), we arrive at the following feature selection constraints:

$$\frac{\sum_{i=1}^m a_{ij} f_{i\hat{k}} x_i}{\sum_{i=1}^m f_{i\hat{k}} x_i} \geq \frac{\sum_{i=1}^m a_{ij} f_{ik} x_i}{\sum_{i=1}^m f_{ik} x_i}, \qquad (10)$$

for all $j \in \mathcal{S}_{\hat{k}}$, $\hat{k}, k = 1 \ldots r$, $\hat{k} \neq k$.

These are fractional 0–1 constraints, and in order to be tackled by industrial optimization solvers, they need to be linearized. The linearization is based on a very simple idea:

Theorem 1 (Wu [1]) *A polynomial mixed 0–1 term $z = xy$, where x is a 0–1 variable, and y is a continuous variable taking any positive value, can be represented by the following linear inequalities: (1) $y - z \leq M - Mx$; (2) $z \leq y$; (3) $z \leq Mx$; (4) $z \geq 0$, where M is a large number greater than y.*

A simple proof of this result can be found in [1].

So, let us introduce variables

$$y_k = \frac{1}{\sum_{i=1}^m f_{ik} x_i},\ k = 1 \ldots r. \qquad (11)$$

Since f_{ik} can take values only zero or one, equation (11) can be equivalently rewritten as

$$\sum_{i=1}^m f_{ik} x_i \geq 1,\ k = 1 \ldots r. \qquad (12)$$

$$\sum_{i=1}^m f_{ik} x_i y_k = 1,\ k = 1 \ldots r. \qquad (13)$$

In terms of the new variables y_k, condition (10) is replaced by

$$\sum_{i=1}^m a_{ij} f_{i\hat{k}} x_i y_{\hat{k}} \geq \sum_{i=1}^m a_{ij} f_{ik} x_i y_k \qquad (14)$$

for all $j \in \mathcal{S}_{\hat{k}}$, $\hat{k}, k = 1 \ldots r$, $\hat{k} \neq k$. Next, observe that the term $x_i y_k$ is present in (14) if and only if $f_{ik} = 1$, i.e., $i \in \mathcal{F}_k$. So, there are totally only m of such products in (14), and hence we can introduce m variables $z_i = x_i y_k$, $i \in \mathcal{F}_k$ to linearize the system by **Theorem 1**. Obviously, the parameter M can be set to 1. So, instead of (13) and (14), we have the following constraints:

$$\sum_{i=1}^m f_{ik} z_i = 1,\ k = 1 \ldots r. \qquad (15)$$

$$\sum_{i=1}^m a_{ij} f_{i\hat{k}} z_i \geq \sum_{i=1}^m a_{ij} f_{ik} z_i \qquad (16)$$

for all $j \in \mathcal{S}_{\hat{k}}$, $\hat{k}, k = 1 \ldots r$, $\hat{k} \neq k$;

$$y_k - z_i \leq 1 - x_i,\ z_i \leq y_k,\ z_i \leq x_i,\ z_i \geq 0, \qquad (17)$$

when $i \in \mathcal{F}_k$.

3 Formulations and Algorithms for Feature Selection

As we mentioned in the introduction, we formulate the feature selection problem as an optimization task and use the objective function either to maximize the class separation or to minimize the information loss. In the latter case the goal is to select as many features as possible with minimum decrease of their weights, so the objective function may be expressed as

$$\max \sum_{i=1}^{m} x_i \tag{18}$$

independently of what type of constraints do we use. Class separation measures are more criterion-specific, so we cannot formulate a unique objective function in this case.

Below, we formally state three optimization formulations used to perform the feature selection, and specify applied solving methods and criteria for classification of test set samples.

3.1 k-means local optimality

The objective (18) was used with constraints (2) applied to the training set. The variables x_i are continuous and bounded as $0 \leq x_i \leq 1$, $i = 1 \ldots m$. This is a Linear Programming formulation, and it can be addressed by a standard software like CPLEX [2].

To perform the test set classification, we choose for each test sample $b = (b_i)_{i=1\ldots m}$ the class $\mathcal{S}_{\hat{k}}$ such that

$$\sum_{i=1}^{m} (b_i - c_{i\hat{k}})^2 x_i \leq \sum_{i=1}^{m} (b_i - c_{ik})^2 x_i \tag{19}$$

for all $k = 1 \ldots r$.

3.2 Pairwise threshold constraints

In this case we applied the class separation objective of the form

$$\max D_{ext} - D_{int} \tag{20}$$

subject to the constraints (3), (4). Again, the variables x_i are continuous and bounded as $0 \leq x_i \leq 1$, $i = 1 \ldots m$, and this is a Linear Programming formulation that can be addressed by a standard software like CPLEX [2]. However, because the pairwise threshold constraints are very strong and require a very distinct separation of classes, there is a good chance that the only feasible solution is trivial: $x = 0$. This may be caused by outliers in the training set or just scattered distribution within the classes. Hence, we have to make certain relaxation of the constraint if a non-trivial solution is not possible.

Fortunately, it can be performed analyzing dual variables corresponding to the trivial solution. Indeed, if the dual variable corresponding to a constraint is nonzero, we know that this constraint is active and keeps the optimal solution from improvement. So, as long as $x = 0$ is the only feasible solution to the problem, we iteratively remove constraints with corresponding nonzero dual variables unless we obtain the opportunity to improve the solution. If this procedure leads to removal of all constraints, we conclude that the given feature selection problem is not suitable for the pairwise threshold constraints.

To perform the test set classification, we apply the nearest neighborhood criterion. That is, for a test sample $b = (b_i)_{i=1\ldots m}$, we find the sample $\hat{j}$ such that

$$\sum_{i=1}^{m} \left(b - a_{i\hat{j}}\right)^2 x_i \leq \sum_{i=1}^{m} (b - a_{ij})^2 x_i \tag{21}$$

for all $j = 1 \ldots n$, and assign b to the class $\mathcal{S}_{\hat{k}} \ni \hat{j}$.

3.3 Consistent biclustering

We have chosen the objective (18) for feature selection via consistent biclustrering. As we mentioned above, the variables x_i are considered to be 0–1 in this case. With the constraints (15)-(17) it forms a linear mixed 0–1 program. Unfortunately, while the linearization by **Theorem 1** works nicely for small-size problems, it often creates instances, where the gap between the integer programming and the linear programming relaxation optimum solutions is very big for larger problems. As a consequence, the instance can not be solved in a reasonable time even with the best techniques implemented in modern integer programming solvers. Hence, we have developed an alternative approach.

Consider the meaning of variables z_i. We have introduced them so that

$$z_i = \frac{x_i}{\sum_{\ell=1}^m f_{\ell k} x_\ell}, \ i \in \mathcal{F}_k. \tag{22}$$

Thus, for $i \in \mathcal{F}_k$, z_i is the reciprocal of the cardinality of the class $\mathcal{F}_k$ after the feature selection, if the i-th feature is selected, and 0 otherwise. This suggests that z_i is also a binary variable by nature as x_i is, but its nonzero value is just not set to 1. It is not known unless the optimal sizes of feature classes are obtained. However, knowing z_i is sufficient to define the value of x_i, and the system of constraints with respect only to the continuous variables $0 \leq z_i \leq 1$ constitutes a linear relaxation of the biclustering constraints (10). Furthermore it can be strengthened by the system of inequalities connecting z_i to x_i. Indeed, if we know that no more than m_k features can be selected for class $\mathcal{F}_k$, then it is valid to impose:

$$x_i \leq m_k z_i, \ x_i \geq z_i, \ i \in \mathcal{F}_k. \tag{23}$$

Hence, we used the following iterative heuristic algorithm for feature selection via consistent biclustering:

Algorithm 1

```
1. Assign $m_k := |\mathcal{F}_k|$, $k = 1 \ldots r$.
2. Solve the mixed 0–1 programming formulation using the inequalities (23) instead of (17).
3. If $m_k = \sum_{i=1}^m f_{ik} x_i$ for all $k = 1 \ldots r$, go to 6.
4. Assign $m_k := \sum_{i=1}^m f_{ik} x_i$ for all $k = 1 \ldots r$.
5. Go to 2.
6. STOP.
```

Another modification that was used to improve the quality of the feature selection is strengthening of the class separation by introduction of a coefficient greater than 1 for the right-hand side of the inequality (10). In this case, we improve (16) by the relation

$$\sum_{i=1}^m a_{ij} f_{i\hat{k}} z_i \geq (1+t) \sum_{i=1}^m a_{ij} f_{ik} z_i \tag{24}$$

for all $j \in \mathcal{S}_{\hat{k}}$, $\hat{k}, k = 1 \ldots r$, $\hat{k} \neq k$, and $t > 0$ is a constant that becomes a parameter of the method. We used $t = 0.1$ for our computational experiments.

After the feature selection is done, we perform classification of test samples according to (9). That is, if $b = (b_i)_{i=1\ldots m}$ is a test sample, we assign it to the class $\mathcal{S}_{\hat{k}}$ satisfying

$$\frac{\sum_{i=1}^m b_i f_{i\hat{k}} x_i}{\sum_{i=1}^m f_{i\hat{k}} x_i} \geq \frac{\sum_{i=1}^m b_i f_{ik} x_i}{\sum_{i=1}^m f_{ik} x_i} \tag{25}$$

for all $k = 1 \ldots r$, $\hat{k} \neq k$.

4 Computational Experiments

4.1 ALL vs. AML data set

We applied our methodology to a well-researched microarray data set containing samples from patients diagnosed with *acute lymphoblastic leukemia* (ALL) and *acute myeloid leukemia* (AML) diseases [3]. It has been the subject of a variety of research papers, e.g. [4, 5, 6, 7]. This data set was also used in the CAMDA 2001 data contest. It is divided into two parts – the training set (27 ALL, 11 AML samples), and the test set (20 ALL, 14 AML samples), and involves 7070 human genes (features).

The feature selection program with k-means local optimality constraints (18),(2) delivered the optimum value 7069.3582 (which means that almost all features were selected with weights close to 1). The subsequent classification of the test set by (19) gave two misclassifications: the AML-sample 64 and AML-sample 66 were classified into the ALL class.

The pairwise threshold program (18),(3),(4) selected 1457 features with nonzero weights. The subsequent classification of the test set was perfect: all ALL and AML test samples were classified into appropriate classes.

The biclustering feature selection **Algorithm 1** selected 3439 features for class ALL and 3242 features for class AML. The subsequent classification by (25) contained only one error: the AML-sample 66 was classified into the ALL class.

To provide justification of the quality of this result, we should mention that the support vector machines (SVM) approach delivers up to 5 classification errors on the ALL vs. AML data set depending on how the parameters of the method are tuned [6]. Furthermore, the perfect classification was obtained only with one specific set of values of the parameters.

4.2 Colon cancer data set

A colon cancer microarray data set including expression profiles of 2000 genes from 22 normal tissues and 40 tumor samples was published in [8]. We randomly selected 11 normal and 20 tumor samples into the training set. The other half of samples were used as the test set.

The feature selection program with k-means local optimality constraints (18),(2) delivered the optimum value 1903.045. The number of features selected with nonzero weights was 1901. The classification errors were as follows: 4 Normal samples (8, 12, 34, 36) are classified into Tumor class, and 2 Tumor samples (30, 36) are classified into Normal class.

The pairwise threshold constraints allowed for a feasible solution only after two iterations of exclusion of active constraints, and after that only 32 features were selected with nonzero weights. The misclassified samples are 5 Normal (2, 8, 12, 34, 36), and 2 Tumor (30, 36).

5 Conclusions

We have developed an optimization framework for handling major data mining problems, which provides a unified methodology for feature selection and classification with the possibility of outlier detection. It has a very natural connection to the conceptions of unsupervised clustering. Since the used unsupervised clustering criteria are not fixed, the methodology is highly flexible and potentially may be used to process data of arbitrary nature. The fact that the practically important data mining problems can be represented as optimization problems allows us to use standard optimization software packages to solve them. This direction gives us a promise for more efficient treatment of real-world problems, whose original formulation is normally quite fuzzy.

The good performance on known microarray data sets confirms reliability of the applied methodology.

References:

[1] T.-H. Wu, A note on a global approach for general 0–1 fractional programming, *European J. Oper. Res.*, Vol. 101, 1997, pp. 220–223.

[2] *CPLEX 9.0 User's Manual.* ILOG Inc., 2004.

[3] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, Vol. 286, 1999, pp. 531–537.

[4] A. Ben-Dor, L. Bruhn, I. Nachman, M. Schummer, and Z. Yakhini, Tissue classification with gene expression profiles, *J. Comput. Biol.*, Vol. 7, 2000, pp. 559–584.

[5] A. Ben-Dor, N. Friedman, and Z. Yakhini, Class discovery in gene expression data, in *Proc. Fifth Annual Inter. Conf. on Computational Molecular Biology (RECOMB)*, 2001.

[6] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, Feature selection for svms, in *Proc. NIPS Conf.*, 2001.

[7] E. Xing and R. Karp, Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts, *Bioinformatics Discovery Note*, Vol. 1, 2001, pp. 1–9.

[8] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, *Proc. Natl. Acad. Sci.*, Vol. 96, 1999, pp. 6745–6750.