ELSEVIER
CrossMark
Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 31 (2014) 58 – 65
Procedia
Computer Science

2nd International Conference on Information Technology and Quantitative Management,
ITQM 2014

# Parallel Regularized Multiple-Criteria Linear Programming

Zhiquan Qi[a], Vassil Alexandrov[b,*], Yong Shi[a], Yingjie Tian[a]

[a] *Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences, Beijing 100190, China*
[b] *ICREA and Barcelona Supercomputing Center, C/Jordi Girona, 29, Edifici Nexus II, E-08034 Barcelona, Spain*

**Abstract**

In this paper, we proposed a new parallel algorithm: Parallel Regularized Multiple-Criteria Linear Programming (PRMCLP) to overcome the computing and storage requirements increased rapidly with the number of training samples. Firstly, we convert RMCLP model into a unconstrained optimization problem, and then split it into several parts, and each part is computed by a single processor. After that, we analyze each part's result for next cycle going. By doing this, we are be able to obtain the final optimization solution of the whole classification problem. All experiments in public datasets show that our method greatly increases the training speed of RMCLP in the help of multiple processors.

*Keywords:* PRMCLP, parallel algorithm, data mining;

## 1. Introduction

Nowadays, the coming of Big Data bring us unprecedented opportunities and challenges. On the one hand, Data is becoming the larger and more complex, which causes us to be puzzled facing the vast ocean of information. However, on the other hand, in order to solve the important management problem, I usually do not gain enough knowledge to support our decision. One of the most important reason is that we still do not have the capabilities to extract useful knowledge from Big Data. As a result, more and more people begin to research new data mining methods and technology to deal with the increasing complex data. In this paper, we focuss on the the research of parallel algorithm based on Regularized Multiple-Criteria Linear Programming (RMCLP)[1] to further accelerate the training speed, which will provide a possible way in order to the big data problem.

Classification is one of the most basic and key problem in machine learning and data mining field, and various classification algorithm have been developed in the last few years[2,3,4,5,6]. Support Vector Machine(SVM)[7,8,9] is one of the most popular methods. However, the idea of applying optimization techniques to solve classification problem can be dated back to more than 70 years ago when linear discriminant analysis (LDA)[10] was first proposed in 1936.[11] has proposed a similar model with SVM using the large margin idea in 1960's. From 1980s to 1990s, Glover proposed

[*] Corresponding author. Tel.: +34(0)934137576 ; fax: +34(0)934137721.
*E-mail address:* vassil.alexandrov@bsc.es

doi:10.1016/j.procs.2014.05.245

a number of linear programming models to solve discriminant problems with a small sample size of data[12,13]. Other classification models also can found in[14,15,16,17,18]. Recently, Shi and his colleagues[19] extend Glover's method into classification via multiple criteria linear programming (MCLP), and then various improved algorithms were proposed one after the other[1,20,21,22]. These mathematical programming approaches to classification have been applied to handle many real world data mining problems, such as credit card portfolio management[23,24,25], bioinformatics[26], information intrusion and detection[27], firm bankruptcy[28,29], and etc.

In order to realize the classification algorithm parallelization, there are usually two strategies as follows: 1) the parallel algorithm is designed by the divide-and-rule tactics. For a large scale problem, we can divide it into several sub-problem, which is mutually independent and have same form with the primal problem. And then, these sub-problems are solved by the recursion way. At last, combining these results, we can obtain the solution of primal problem. We can find typical methods in[30,31,32]. 2) parallelling the serial algorithm. This strategy tries to find the parallel nature of the algorithm itself, and finally finish the classification's parallelization. several typical methods include[33,34,35,36].

In this paper, we focus on the research of RMCLP, and propose a Parallel version of RMCLP algorithm (PRMCLP) in order to overcome the compute and storage requirements increase rapidly with the number of training sample. Inspire by[37], we adopt the second strategy to parallelize our algorithm.

Firstly, we convert RMCLP model into a unconstrained optimization problem, and then split it into several parts, which is performed in $p$ processors at the same time. After that, we summarize and analyze the results obtained by each processors, and take them as the parameter input of each sub-problem in next step. Keeping the cycle going, we will obtain the optimization solution of the whole classification problem until satisfying the terminal condition. Experiments in public datasets show that our method greatly increases the training speed of RMCLP in the help of $p$ processors.

The remaining parts of the paper are organized as follows. Section 2 introduces the basic formulation of MCLP and RMCLP; Section 3 describes in detail our proposed Algorithms PRMCLP; All experiment results are shown in the section 4; Last section gives the conclusions.

## 2. Regularized MCLP for Machine Learning

We give a brief introduction of MCLP in the following. For classification about the training data

$$T = \{(x_1, y_1), \cdots, (x_l, y_l)\} \in (\Re^n \times \mathcal{Y})^l, \tag{1}$$

where $x_i \in \Re^n, y_i \in \mathcal{Y} = \{1, -1\}, i = 1, \cdots, l$, data separation can be achieved by two opposite objectives. The first objective separates the observations by minimizing the sum of the deviations (MSD) among the observations. The second maximizes the minimum distances (MMD) of observations from the critical value[13]. The overlapping of data $\xi^{(1)}$ should be minimized while the distance $\xi^{(2)}$ has to be maximized. However, it is difficult for traditional linear programming to optimize MMD and MSD simultaneously. According to the concept of Pareto optimality, we can seek the best trade-off of the two measurements[24,25]. So MCLP model can be described as follows:

$$\begin{aligned}
\min_{\xi^{(1)}} \quad & e^\top \xi^{(1)} \ \& \ \max_{\xi^{(2)}} \quad e^\top \xi^{(2)} \\
\text{s.t.} \quad & (w \cdot x_i) + (\xi_i^{(1)} - \xi_i^{(2)}) = b, \ \text{for } \{i | y_i = 1\}, \\
& (w \cdot x_i) - (\xi_i^{(1)} - \xi_i^{(2)}) = b, \ \text{for } \{i | y_i = -1\}, \\
& \xi^{(1)}, \xi^{(2)} \geq 0,
\end{aligned} \tag{2}$$

where $e \in R^l$ be vector whose all elements are 1, $w$ and $b$ are unrestricted, $\xi_i^{(1)}$ is the overlapping and $\xi_i^{(2)}$ the distance from the training sample $x_i$ to the discriminator $(w \cdot x_i) = b$ (classification separating hyperplane). By introducing penalty parameter $C, D > 0$, MCLP has the following version

$$\begin{aligned}
\min_{\xi^{(1)}, \xi^{(2)}} \quad & Ce^\top \xi^{(1)} - De^\top \xi^{(2)}, \\
\text{s.t.} \quad & (w \cdot x_i) + (\xi_i^{(1)} - \xi_i^{(2)}) = b, \text{for } \{i | y_i = 1\}, \\
& (w \cdot x_i) - (\xi_i^{(1)} - \xi_i^{(2)}) = b, \text{for } \{i | y_i = -1\}, \\
& \xi^{(1)}, \xi^{(2)} \geq 0,
\end{aligned} \tag{3}$$

A lot of empirical studies have shown that MCLP is a powerful tool for classification. However, we cannot ensure this model always has a solution under different kinds of training samples. To ensure the existence of solution, recently, Shi et.al proposed a RMCLP model by adding two regularized items $\frac{1}{2}w^\top Hw$ and $\frac{1}{2}\xi^{(1)\top}Q\xi^{(1)}$ on MCLP as follows (more theoretical explanation of this model can be found in[1]:

$$\begin{array}{ll} \min\limits_{z} & \frac{1}{2}w^\top Hw + \frac{1}{2}\xi^{(1)\top}Q\xi^{(1)} + \frac{1}{2}b^2 + Ce^\top\xi^{(1)} - De^\top\xi^{(2)}, \\ \text{s.t.} & (w\cdot x_i) + (\xi_i^{(1)} - \xi_i^{(2)}) = b, \ \ \text{for } \{i|y_i = 1\}, \\ & (w\cdot x_i) - (\xi_i^{(1)} - \xi_i^{(2)}) = b, \ \ \text{for } \{i|y_i = -1\}, \\ & \xi^{(1)}, \xi^{(2)} \geq 0, \end{array} \tag{4}$$

where $z = (w^\top, \xi^{(1)^\top}, \xi^{(2)^\top}, b)^\top \in R^{n+l+l+1}$, $H \in R^{n\times n}$, is symmetric positive definite matrices. Obviously, the regularized MCLP is a convex quadratic programming. According to dual theorem, (4)~(4) can be turned into

$$\begin{array}{ll} \min\limits_{\alpha,\ \xi^{(1)}} & \frac{1}{2}\alpha^\top(K(A,A^\top) + ee^\top)\alpha + \frac{1}{2}\xi^{(1)^\top}Q\xi^{(1)}, \\ \text{s.t.} & -Q\xi^{(1)} - Ce \leq E\alpha \leq -De, \end{array} \tag{5}$$

where $A = [x_1^\top, \cdots, x_l^\top]^\top \in R^{l\times n}$, $E = diag\{y_1, \cdots, y_l\}$, and $K(A, A^\top) = \Phi(A)\Phi(A)^\top = (\Phi(A)\cdot\Phi(A))_{l\times l}$, and $\Phi$ is a mapping from the input space $R^n$ to some Hilbert space $\mathcal{H}$[38].

Compared with traditional SVM , we can find that the RMCLP model is similar to the Support Vector Machine SVM model in terms of the formation by considering minimization of overlapping of the data. However, RMCLP tries to measure all possible distances $\xi^{(2)}$ from the training samples $x_i$ to the separating hyperplane, while SVM fixes the distance as 1 (through bounding planes $(w\cdot x) - b = \pm 1$) from the support vectors. Although the interpretation can vary, RMCLP addresses more control parameters than the SVM, which may provide more flexibility for better separation of data under the framework of the mathematical programming. In addition, different from the traditional SVM, the RMCLP considers all the samples to solve the classification problem and thus is insensitive to outliers.

## 3. Parallel RMCLP

### 3.1. Algorithm Structure

In order to realize the parallelization of RMCLP, we firstly translate RMCLP into a unconstrained optimization problem. To simplify, (5) can be rewritten as

$$\begin{array}{ll} \min\limits_{\pi} & \frac{1}{2}\pi^\top\Lambda\pi, \\ \text{s.t.} & G\pi - Ce \leq 0, \\ & H\pi + De \leq 0, \end{array} \tag{6}$$

where $\pi = [\alpha^\top,\ \xi^{(1)^\top}]^\top$, and $G = [-Q,\ -E]$, $H = [E,\ O]$, $O \in R^{l\times l}$ is a null matrix, $\Lambda$ is written as

$$\begin{pmatrix} K(A,A^\top) + ee^\top & 0 \\ 0 & Q \end{pmatrix}. \tag{7}$$

Next, we turn the objective (5) into the following unconstrained optimization problem

$$\min_\pi f(\pi) \ = \frac{1}{2}\pi^\top\Lambda\pi + \lambda^\top\max\{G\pi - Ce, 0\}^2 + \mu^\top\max\{H\pi + De, 0\}^2, \tag{8}$$

where $C, D \in R$ are the artificial parameters, and $\lambda = \{\lambda_1, \cdots, \lambda_l\}$, $\mu = \{\mu_1, \cdots, \mu_l\}$. Define $d$ is the search direction of the optimization problem (8), here, we choose the negative gradient direction as the feasible direction:

$$d = -\nabla f(\pi)/||\nabla f(\pi)||, \tag{9}$$

where

$$\nabla f(\pi) = \Lambda\pi + 2\lambda^\top diag(G^\top\max\{G\pi - Ce, 0\}) + 2\mu^\top diag(H^\top\max\{H\pi + De, 0\}). \tag{10}$$

Now, we use PVD idea to split our model[37]. Suppose we can use $p$ processor, the variable $\pi$ of the unconstrained optimization problem (8) can be divided into $p$ chunks: $\{\pi_1, \cdots, \pi_p\}$, where the dimension of the $i$th chunk is $m_i$, i.e.,

$$\pi = \{\pi_1, \cdots, \pi_m\},\ \pi_i \in R^{m_i},\ i = 1, \cdots, p,\ \sum_{i=1}^{p} m_i = 2l. \tag{11}$$

In the next step, we allocate the $p$ variable to $p$ processor, and decompose the problem (8) into the subproblem with $m_i$ dimension. Each processor solves one corresponding subproblem, which update other variables on the basis of some rules except for computing the $m_i$ variables itself. After each processor finishes updating, we will perform a quick synchronous step: searching the results obtained by each computing unit and compute the current solution. Repeating the course, our algorithm can be described as

**Algorithm 1** Parallel RMCLP

```
step 1: Given the initial point $\pi^0$, and penalty factors $\lambda, \mu > 0$. $k = 0$, and let $\pi^0 = \{\pi_1^0, \cdots, \pi_p^0\}$, $\lambda = \{\lambda_1, \cdots, \lambda_p\}$,
        $\mu = \{\mu_1, \cdots, \mu_p\}$, kernel matrix $\Lambda$ is divided into $\{\Lambda_1, \cdots, \Lambda_p\}$, similarly $G = \{G_1, \cdots, G_p\}$, $H = \{H_1, \cdots, H_p\}$,
        and storage $\pi_i^k, \lambda_i, \mu_i$ in the $i$th processor, $i = 1, \cdots, p$.
Step 2: Computing for each processor, $i = 1, \cdots, p$:
          $p_i^1 = \Lambda_i \pi_i^k$,
          $p_i^2 = \lambda_i^\top diag(G_i^\top \max\{G_i \pi_i^k - Ce, 0\})$,
          $p_i^3 = \mu^\top diag(H_i^\top \max\{H_i \pi_i^k + De, 0\})$.
Step 3: Computing:
          $d^k = \sum_{i=1}^{p} p_i^1 + 2\sum_{i=1}^{p} p_i^2 + 2\sum_{i=1}^{p} p_i^3$,
          If $d^k = 0$, the optimization solution is $\pi^k$, go to Step 5, else $d^k = d^k/||d^k||$.
Step 4: Computing for each processor, $i = 1, \cdots, p$:
          $\min_{\pi_i, \nu_i} \omega_i^k(\pi_i, \nu_i) = f(\pi_i, \pi_{\bar{i}}^k + diag(d_{\bar{i}}^k)\nu_i)$, where $\bar{i}$ is the complementary set of $i$, and the approximate solution
          $(\pi_i^k, \nu_i^k) \in R^{m_i} \times R^{p-1}$, let $\pi^{k_i} = \{\pi_i^k, \pi_{\bar{i}}^k + \nu_i diag(d_{\bar{i}}^k)\}$.
Step 5: compute $\tau_0, \tau_1, \cdots, \tau_p$, and solve
          $\min_{\tau_0, \tau_1, \cdots, \tau_p} f(\tau_0 \pi^k + \sum_{i=1}^{p} \tau_i \pi^{k_i}),\ s.t. \sum_{i=0}^{p} \tau_i = 1$,
          let $\pi^{k+1} = \tau_0 \pi^k + \sum_{i=1}^{p} \tau_i \pi^{k_i},\ k = k + 1$, if $\nabla f(\pi^{k+1}) < \varepsilon$, terminate the algorithm, else go to step 2, where $\varepsilon$ is small
          enough positive number.
```

### 3.2. Convergence Analysis

**Theorem 3.1.** *The sequence generated by* $\{\pi^k\}$ *of* **Algorithm 1** *either terminates at a stationary point* $\{\pi^{\bar{k}}\}$*, or is an infinite sequence, whose accumulation point is stationary and* $\lim_{k\to\infty} \nabla f(\pi^k) = 0$.

**Proof.** For $\forall \pi, \pi' \in R^{2l}$, we have

$$\begin{aligned}\nabla f(\pi) = {} & \Lambda\pi + 2\lambda^\top diag(G^\top \max\{G\pi - Ce, 0\}) \\ & +2\mu^\top diag(H^\top \max\{H\pi + De, 0\}).\end{aligned} \tag{12}$$

So

$$\begin{aligned} & ||\nabla f(\pi) - \nabla f(\pi')|| \\ = {} & ||\Lambda(\pi - \pi') + 2\lambda^\top diag(G^\top(\max\{G\pi - Ce, 0\} - \max\{G\pi' - Ce, 0\})) \\ & +2\mu^\top diag(H^\top(\max\{H\pi + De, 0\} - \max\{H\pi' + De, 0\}))|| \\ \le {} & ||\Lambda||||(\pi - \pi')|| + 2||\lambda^\top||||diag(G^\top(\max\{G\pi - Ce, 0\} - \max\{G\pi' - Ce, 0\}))|| \\ & +2||\mu^\top||||diag(H^\top(\max\{H\pi + De, 0\} - \max\{H\pi' + De, 0\}))||.\end{aligned} \tag{13}$$

We can prove easily

$$\begin{aligned} & diag(G^\top(\max\{G\pi - Ce, 0\} - \max\{G\pi' - Ce, 0\})) \le G^\top(\pi - \pi') \\ & diag(H^\top(\max\{H\pi + De, 0\} - \max\{H\pi' + De, 0\})) \le H^\top(\pi - \pi').\end{aligned} \tag{14}$$

Let $||\Lambda|| + 2||\Lambda||||G^\top|| + 2||\mu^\top||||H^\top|| = K$, we obtain

$$||\nabla f(\pi) - \nabla f(\pi^{'})|| \leq K(\pi - \pi^{'}). \tag{15}$$

As the result, according to the **Theorem 2.2** in[39], $\{\pi^k\}$ either terminates at a stationary point $\pi^{\bar{k}}$, or is an infinite sequence, whose accumulation point is stationary and $\lim\limits_{k\to\infty} \nabla f(\pi^k) = 0$. □

**Theorem 3.2.** *If* $\Lambda$ *of* **Algorithm 1** *is positive definite, then the sequence of iterates* $\{\pi^k\}$ *generated by the subproblem of (8)converges linearly to the unique solution* $\bar{\pi}$ *, and the rate of convergence is*

$$||\pi^k - \bar{\pi}|| \leq (\frac{2}{\gamma}(f(\pi^k) - f(\bar{\pi})))^{\frac{1}{2}}(1 - \frac{1}{p}(\frac{\gamma}{K})^2)^{\frac{1}{2}}, \tag{16}$$

*where* $\gamma, K > 0$ *are the constants.*

***Proof.*** *For* $\forall \pi, \pi^{'} \in R^{2l}$*, we have*

$$\begin{aligned} &(\nabla f(\pi) - \nabla f(\pi^{'}))(\pi - \pi^{'}) \\ &= (\pi - \pi^{'})^\top \Lambda(\pi - \pi^{'}) + (2\lambda^\top diag(G^\top(\max\{G\pi - Ce, 0\} - \max\{G\pi^{'} - Ce, 0\})) \\ &\quad +2\mu^\top diag(H^\top(\max\{H\pi + De, 0\} - \max\{H\pi^{'} + De, 0\}))))(\pi - \pi^{'}). \end{aligned} \tag{17}$$

*We easily know that*

$$\begin{aligned} &diag(G^\top(\max\{G\pi - Ce, 0\} - \max\{G\pi^{'} - Ce, 0\}))(\pi - \pi^{'}) \geq 0, \\ &diag(G^\top(\max\{G\pi - Ce, 0\} - \max\{G\pi^{'} - Ce, 0\}))(\pi - \pi^{'}) \geq 0. \end{aligned} \tag{18}$$

*Since* $\Lambda$ *is a positive definite matrix, we have*

$$(\nabla f(\pi) - \nabla f(\pi^{'}))(\pi - \pi^{'}) \geq (\pi - \pi^{'})^\top \Lambda(\pi - \pi^{'}) \geq \frac{\gamma}{2}||\pi - \pi^{'}||^2, \forall \pi \in R^{2l}, \tag{19}$$

*where* $\gamma$ *is a constant. As the result, subproblem of (8)converges linearly to the unique solution* $\bar{\pi}$ *, and the rate of convergence is*

$$||\pi^k - \bar{\pi}|| \leq (\frac{2}{\gamma}(f(\pi^k) - f(\bar{\pi})))^{\frac{1}{2}}(1 - \frac{1}{p}(\frac{\gamma}{K})^2)^{\frac{1}{2}}, \tag{20}$$

□

## 4. Numerical Experiment

Our algorithm code was programmed in MATLAB 2010. The experiment environment: Intel Core I5 CPU, 2 GB memory. The ”fminbnd” and ”quadprog” function with MATLAB is employed to solve quadratic programming problem related to this paper.

To demonstrate the capabilities of our algorithm, we report results on MNIST data sets and UCI data sets, respectively. In all experiments, our method is compared under different CPU processors.

The testing accuracies are computed using standard 10-fold cross validation ([8]). The parameter $C$ and the RBF kernel parameter $\sigma$ are selected from the set $\{2^i|i = -7, \cdots, 7\}((C, D)$ cross validation on the tuning set comprising of random 10% of the training data. Once the parameters are selected, the tuning set was returned to the training set to learn the final decision function.

### 4.1. MNIST Dataset

MNIST Dataset is a handwritten digit dataset with samples from ‘0’ to ‘9’. The size of each sample is 16 × 16 pixels. The same as the literature[40], we test on the ‘5’ vs ‘8’ classification problem in the case of linear kernel. The results are showed in Table 1.

From Table 1, we can find that the accuracy about handwritten digit dataset is the same in the case of different processors, and the training time is greatly deduced with the increase of processors. This shows that our parallel algorithm is very effective.

Table 1. PRMCLP's Training time for '5' and '8' datasets on different sample numbers and processors.

| Method | Training subset size | | | | |
|---|---|---|---|---|---|
| | 200 | 400 | 600 | 800 | 1000 |
| accuracy | 96.17± 1.27 | 400 96.28±1.02 | 97.13 ± 1.34 | 97.56±1.69 | 98.24±0.89 |
| 1 procs(second) | 8.21 | 28.49 | 43.20 | 67.21 | 76.38 |
| 2 procs | 4.39 | 14.58 | 22.12 | 34.65 | 40.45 |
| 4 procs | 2.14 | 7.24 | 13.33 | 18.89 | 22.43 |
| 6 procs | 1.44 | 4.92 | 6.90 | 11.32 | 14.17 |

### 4.2. *UCI Datasets*

In the section, we use respectively Sonar, Ionosphere, Australian, Pima-Indian, CMC, Votes, WPBC to estimate our methods(Table 2 give the description about these data sets).

Table 2. Description of UCI data sets

| datasets | ♯ examples (L) | ♯ dimension (N) |
|---|---|---|
| Sonar | 208 | 60 |
| Ionosphere | 351 | 34 |
| Australian | 690 | 14 |
| Pima-Indian | 768 | 8 |
| CMC | 1473 | 9 |
| Votes | 435 | 16 |
| WPBC | 110 | 32 |

The Table 3 give the experiments results in the case of RBF kernel.

Table 3. PRMCLP's training time on UCI data sets

| Dataset | accuracy | 1 procs | 2 procs | 4 procs | 6 procs |
|---|---|---|---|---|---|
| Sonar | 78.21± 4.46 | 46.35 | 24.18 | 13.62 | 14.12 |
| Ionosphere | 87.22± 6.45 | 148.12 | 76.54 | 38.75 | 25.34 |
| Australian | 86.34 ± 4.23 | 284.76 | 143.52 | 72.32 | 49.77 |
| Pima-Indian | 78.12 ± 5.45 | 331.34 | 169.31 | 87.62 | 56.61 |
| CMC | 70.18 ± 3.69 | 605.17 | 310.28 | 160.23 | 108.12 |
| Votes | 95.54 ± 3.48 | 198.23 | 101.01 | 54.29 | 34.52 |
| WPBC | 82.75 ± 2.92 | 22.25 | 11.57 | 5.76 | 3.43 |

From Table 3, we can find that PRMCLP's the training time in the condition of 10-fold cross validation consumed based on multiple processors is much less than ones based on single processor while their accuracy are in the same level. with the increase of processors, the computing speed has a great improvement, which fully shows our algorithm has a good parallelism.

## 5. Conclusion

In this paper, a new parallel algorithm: Parallel Regularized Multiple-Criteria Linear Programming (PRMCLP) was proposed. With the help of multiple processors, the performance of PRMCLP in public datasets has a great improvement. All experiments show our method's effectiveness, In the future work, in order to deal with big data, how to further accelerate our algorithm is under our consideration.In addition,the extension to semi-supervised learning and multi-class classification is also interesting.

## Acknowledgements

This work has been partially supported by China Postdoctoral Science Foundation under Grant No.2013M530702, and grants from National Natural Science Foundation of China(NO.11271361), key project of National Natural Science Foundation of China(NO.71331005), Major International (Regional) Joint Research Project(NO.71110107026), and the Ministry of water resources' special funds for scientific research on public causes (No. 201301094).

## References

1. Shi, Y., Tian, Y., Chen, X., Zhang, P.. Regularized multiple criteria linear programs for classification. *Science in China Series F: Information Sciences* 2009;**52**(10):1812–1820.
2. Qi, Z., Xu, Y., Wang, L., Song, Y.. Online multiple instance boosting for object detection. *Neurocomputing* 2011;**74**:1769–1775.
3. Qi, Z., Tian, Y., Shi, Y.. Twin support vector machine with universum data. *Neural Networks* 2012;**36C**:112–119. doi:doi: 10.1016/j.neunet.2012.09.004.
4. Qi, Z., Tian, Y., Shi, Y.. Robust twin support vector machine for pattern classification. *Pattern Recognition* 2012;**46(1)**:305–316.
5. Qi, Z., Tian, Y., Shi, Y.. Laplacian twin support vector machine for semi-supervised classification. *Neural Networks* 2012;**35**:46–53.
6. Qi, Z., Tian, Y., Shi, Y.. Structural twin support vector machine for classification. *Knowledge-Based Systems* 2013;doi:DOI: 10.1016/j.knosys.2013.01.008.
7. Vapnik, . *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc.; 1995. ISBN 0-387-94559-8.
8. Deng, N., Tian, Y.. *Support vector machines: Theory, Algorithms and Extensions*. Science Press, China; 2009.
9. Trafalis, T.B., Gilbert, R.C.. Robust classification and regression using support vector machines. *European Journal of Operational Research* 2006;**173**(3):893–909.
10. Fisher, R.A.. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* 1936;**7**:179–188.
11. Mangasarian, O.L., Musicant, D.R.. Generalized support vector machines. In: *Advances in Large Margin Classifiers*. 2000, p. 135–146.
12. Freed, N., Glover, F.. Simple but powerful goal programming models for discriminant problems. *European Journal of Operational Research* 1981;**7**(1):44–60.
13. Freed, N., Glover, F.. Evaluating alternative linear programming models to solve the two-group discriminant problem. *Decision Science* 1986;**17**:151–162.
14. Amo, A., Montero, J., Biging, G., Cutello, V.. Fuzzy classification systems. *European Journal of Operational Research* 2004;**156**(2):495–507.
15. Krzysko, M., Wolynski, W.. New variants of pairwise classification. *European Journal of Operational Research* 2009;**199**(2):512–519.
16. Uney, F., Turkay, M.. A mixed-integer programming approach to multi-class data classification problem. *European Journal of Operational Research* 2006;**173**(3):910–920.
17. Li, D., Glazebrook, K.D.. A bayesian approach to the triage problem with imperfect classification. *European Journal of Operational Research* 2011;**215**(1):169–180.
18. Pasiouras, F., Gaganis, C., Zopounidis, C.. Multicriteria classification models for the identification of targets and acquirers in the asian banking sector. *European Journal of Operational Research* 2010;**204**(2):328–335.
19. Olson, D., Shi, Y.. *Introduction to business data mining*. Irwin-McGraw-Hill Series: Operations and Decision Sciences. McGraw-Hill; 2006. ISBN 9780071244701.
20. Kou, G., Shi, Y., Wang, S.. Multiple criteria decision making and decision support systems - guest editor's introduction. *Decision Support Systems* 2011;**51**:247–249.
21. Zhang, J., Shi, Y., Zhang, P.. Several multi-criteria programming methods for classification. *Computers and Operations Research* 2009; **36**:823–836.
22. Shi, Y., Lee, H.. A binary integer linear program with multi-criteria and multi-constraint levels. *Computers & Operations Research* 1997; **24**(3):259–273.
23. Kou, G., Peng, Y., Shi, Y., Wise, M., Xu, W.. Discovering Credit Cardholders' Behavior by Multiple Criteria Linear Programming. *Annals of Operations Research* 2005;**135**(1):261–274.
24. Shi, Y., Wise, W., Lou, M.. Multiple criteria decision making in credit card portfolio management. Multiple Criteria Decision Making in New Millennium; 2001, p. 427–436.
25. Shi, Y., Peng, Y., Xu, W., Tang, X.. Data Mining via Multiple Criteria Linear Programming: Applications in Credit Card Portfolio Management. *International Journal of Information Technology and Decision Making* 2002;**1**(1):131–152.
26. Zhang, J., Zhuang, W., Yan, N.. Classification of hiv-1 mediated neuronal dendritic and synaptic damage using multiple criteria linear programming. *Neuroinformatics* 2004;**2**:303–326.
27. Kou, G., Peng, Y., Shi, Y., Chen, Z., Chen, X.. A multiple-criteria quadratic programming approach to network intrusion detection. In: *CASDMKM*. 2004, p. 145–153.
28. Kwak, W., Shi, Y., Eldridge, S.W., Kou, G.. Bankruptcy prediction for japanese firms: using multiple criteria linear programming data mining approach. *International Journal of Business Intelligence and Data Mining* 2006;**1**(4):401–416.
29. He, J., Zhang, Y., Shi, Y., Huang, G.. Domain-driven classification based on multiple criteria and multiple constraint-level programming for intelligent credit scoring. *IEEE Transactions on Knowledge and Data Engineering* 2010;**22**(6):826–838.
30. Graf, H.P., Cosatto, E., Bottou, L., Dourdanovic, I., Vapnik, V.. Parallel support vector machines: The cascade svm. In: *Advances in neural information processing systems*. 2004, p. 521–528.

31. Dong, J.x., Krzyzak, A., Suen, C.Y.. Fast svm training algorithm with decomposition on very large data sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2005;**27**(4):603–618.
32. Dong, J.X., Suen, C.Y., KRZYŻAK, A.. A fast svm training algorithm. *International Journal of Pattern Recognition and Artificial Intelligence* 2003;**17**(03):367–384.
33. Zanghirati, G., Zanni, L.. A parallel solver for large quadratic programs in training support vector machines. *Parallel computing* 2003; **29**(4):535–551.
34. Cao, L.J., Keerthi, S.S., Ong, C.J., Zhang, J., Periyathamby, U., Fu, X.J., et al. Parallel sequential minimal optimization for the training of support vector machines. *Neural Networks, IEEE Transactions on* 2006;**17**(4):1039–1049.
35. Tveit, A., Engum, H.. Parallelization of the incremental proximal support vector machine classifier using a heap-based tree topology. In: *Parallel and Distributed Computing for Machine Learning. The Proceedings of the 14th European Conference on Machine Learning and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2003), Cavtat-Dubrovnik, Cratia*. Citeseer; 2003, .
36. Chang, E.Y.. Psvm: Parallelizing support vector machines on distributed computers. In: *Foundations of Large-Scale Multimedia Information Management and Retrieval*. Springer; 2011, p. 213–230.
37. Mangasarian, L.. Parallel gradient distribution in unconstrained optimization. *SIAM Journal on Control and Optimization* 1995;**33**(6):1916–1925.
38. Chen, W., Tian, Y.. Kernel regularized multiple criteria linear programming. pages 345–352. In: *3rd International Symposium on Optimization and Systems Biology*. 2009, .
39. Ferris, M.C., Mangasarian, O.L.. Parallel variable distribution. *SIAM Journal on Optimization* 1994;**4**(4):815–832.
40. Weston, J., Collobert, R., Sinz, F., Bottou, L., Vapnik, V.. Inference with the Universum. In: *ICML '06: Proceedings of the 23rd international conference on machine learning*. ACM; 2006, p. 1009–1016.