International Journal of Computational Intelligence Systems, Vol. 7, No. 1 (February 2014), 12-24

A DC programming approach for feature selection in the Minimax Probability Machine

Liming Yang *

College of Science, China Agricultural University
Beijing, 100083, China
cauylm@126.com

Ribo Ju
College of Science, China Agricultural University
Beijing, 100083, China
juribo123@163.com

Received 5 May 2011

Accepted 21 July 2013

Abstract

This paper presents a new feature selection framework based on the L_0-norm, in which data are summarized by their moments of the class conditional densities. However, discontinuity of the L_0-norm makes it difficult to find the optimal solution. We apply a proper approximation of the L_0-norm and a bound on the misclassification probability involving the mean and covariance of the dataset, to derive a robust difference of convex functions (DC) program formulation, while the DC optimization algorithm is used to solve the problem effectively. Furthermore, a kernelized version of this problem is also presented in this work. Experimental results on both real and synthetic datasets show that the proposed formulations can select fewer features than the traditional Minimax Probability Machine and the L_1-norm state.

Keywords: Feature selection, Minimax probability machine, DC programming.

1. Introduction

Feature selection for classifiers is an important research tool with many applications[1][2][3] in the machine learning field. Feature selection can be used as a process to reduce the data dimensions for classifications, the objective of which is 2-fold: to select a small feature subset while maintaining high classification accuracy. In this paper, we develop an efficient feature selection method to discriminate between two classes with the data summarized by its moments.

Specifically, given the dataset $D = \{(x_i, y_i | x_i \in R^n, y_i = \pm 1, i = 1, \dots m)$, finding a subset of features for a linear classifier $f(x) = sgn(w^T x - b)$ is equivalent to searching for a sparse weight vector w such that most of the elements of w are zero. This implies that when the ith component of w is zero, the ith component of the observation vector x is irrelevant in deciding the class of x. The L_0-norm of vector w, $\|w\|_0 = card\{i|w_i \neq 0\}$, is defined as the number of nonzero elements in vector w. Thus, L_0-norm offers a significant advantage: it is an ideal approach for enforcing sparsity without sacrificing classifica-

*Address: College of Science, China Agricultural University. Correspondence should be addressed to cauylm@126.com.

tion performance. As a result, feature selection for classification problems can be posed as

$$\begin{aligned} &min \qquad \|w\|_0 &(1)\\ &s.t. \quad some\ classification\ framework. &(2)\end{aligned}$$

The L_0-norm is discontinuous and nonconvex, resulting in an NP-hard optimization in general.

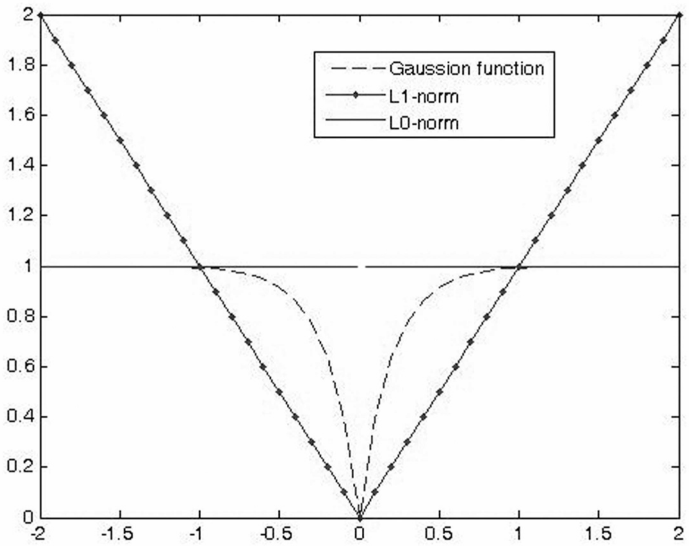

Fig.1 Approximations to zero-norm for the gaussion function η(x) and 1-norm respectively

Feature selection using the moments of class conditional densities has been investigated suing quadratic parabolic interpolation algorithm (S-MPM)[2] and the approximation algorithm based on the L_1-norm (called the L_1-MPM)[4]. The S-MPM is implemented by incorporating L_1-norm into the objective function and the problem is posed as a fractional program[2]. However, thus far, there have been no reports on solving fractional programs effectively. The L_1-MPM is formulated by minimizing the L_1-norm in probability constraints, while the problem is posed as a second-order cone program (SOCP)[5], which can be efficiently solved by interior point codes. However, the L_1-norm minimization criterion is not ideal for feature selection since it only involves minimizing the values of components of weight vector w. In other words, minimizing the L_1-norm generates many components that are close to zero, but not exactly equal to zero.

Ideally, the L_0-norm is the most suitable form for inducing the sparsest classifier since minimizing the L_0-norm of the vector w is nothing other than minimizing the number of representative features in classifier $f(x) = sgn(w^T x + b)$. Nevertheless, the minimization of the L_0-norm is an NP-hard problem[4]. Therefore, most of the effort in feature selection problems has focused on efficient approximation of the L_0-norm; the L_1-norm is only a convex approximation of the L_0-norm (see Fig.1). Thus crucial questions for feature selection include how to approximate the zero-norm effectively and which computational method to use for solving the resulting optimization problem.

Our investigation in this paper is motivated by the following observations:

- The Minimax Probability Machine (MPM)[6][7] has several advantages over other methods in machine learning. It utilizes the mean and covariance of each class of data to find a decision hyperplane, the main benefits of which are that the MPM makes no assumption about the data distribution and has an explicit lower bound on prediction accuracy. Compared with the popular support vector machine (SVM) [8], the MPM has the advantage of using information from the data and can directly yield the probability output for each class of data. However, the MPM formulation does not explicitly combine feature selection and generalization of the model, and thus, it is hard to control the number of selected features.
- Minimizing the L_0-norm is the best way of obtaining a sparse classifier. However, this involves combinatorial optimization, which makes it difficult to find the optimal solution.
- Difference of convex function (DC) programming and the DC optimization algorithm (DCA)[9][10][11][12][13] have been proved to be more robust and more efficient than related standard methods in solving nonconvex and nonsmooth problems.

In this work, we approximate the L_0-norm in a nonconvex way such that the resulting feature selection framework can be formulated as a DC program. The main contributions of this work are as follows:

- By applying the moments of the each class dataset, we propose a new feature selection framework based on the L_0-norm.

- The proposed formulation can be reformulated as a DC program and efficiently solved using the DCA. In addition, we only require that a single SOCP is solved in each iteration.

- We also show how to exploit Mercer kernels in this setting to obtain a nonlinear version.

Throughout the paper we adopt the following notations. The subdifferential of a convex function f is denoted by $\partial f(x)$. A arbitrary dimension vector of ones is denoted by e and $|\cdot|$ denotes absolute value. The base of the natural logarithm will be denoted by ε and ε^{-x} will denote a vector $x \in R^n$ with components ε^{-x_i}.

The rest of this paper is organized as follows. Section 2 gives a short summary of the MPM and DC programming. In Section 3, we propose two new feature selection formulations for classifiers based on L_0-norm . Experimental results for proposed method are shown in Section 4. The concluding section summarizes the main contributions and future directions.

2. Background

2.1. Minimax Probability Machine

The MPM separates two classes of data using the means and covariance of the dataset. The following is a simplified explanation of MPM. A more detailed description can be found in Ref.6. Let X_1 and X_2 denote n dimensional random vectors in a binary classification problem, with mean vectors and covariance matrices given by $X_1 \sim (\mu_1, \Sigma_1)$ and $X_2 \sim (\mu_2, \Sigma_2)$, respectively, where $\mu_1, \mu_2 \in R^n$ and $\Sigma_1, \Sigma_2 \in R^{n\times n}$. Note that both the matrices Σ_1 and Σ_2 are positive semi-definite. The objective of MPM is to formulate a hyperplane $H(w,b) = \{x|w^T x = b\}$ which separates the two classes of samples with maximal probability with respect to all distributions that have these mean and covariance matrices. This is expressed as

$$\begin{aligned} \max \quad & \theta && (3)\\ \text{s.t.} \quad & P\{X_1 \in H_1\} \geqslant \theta && (4)\\ & P\{(X_2 \in H_2\} \geqslant \theta && (5) \end{aligned}$$

where θ represents the lower bounds of the classification probability for future data. Applying the Chebychev Cantelli inequality [14], the problem is expressed as a SOCP and is solved using the efficient interior point algorithm . However, this paradigm does not automatically control the balance between prediction accuracy and the number of selected features.

2.2. DC programming

DC programming and DCA, introduced by Pham Dinh Tao in 1985, constitute the backbone of nonconvex continuous programming. Generally speaking, a DC program takes the form

$$\inf\{f(x) = g(x) - h(x), x \in R^n\} \qquad (P_{dc}) \qquad (6)$$

where g and h are lower semicontinuous proper convex functions on R^n. Such a function f is called a DC function, and g and h are the DC components of f. We use $g^*(y) = \sup\{x^T y - g(x), x \in R^n\}$ to denote the conjugate function of g. The Fenchel-Rockafellar dual of (P_{dc}) is defined as

$$\inf\{h^*(y) - g^*(y), y \in R^n\} \qquad (D_{dc}) \qquad (7)$$

A point x^* that satisfies the following generalized Kuhn-Tucker condition is called a critical point of (P_{dc})

$$\partial h(x^*) \cap \partial g(x^*) \neq \emptyset \qquad (8)$$

The necessary local optimality condition of (P_{dc}) is

$$\partial h(x^*) \subset \partial g(x^*) \neq \emptyset \qquad (9)$$

DCA is an iterative algorithm based on local optimality conditions and duality. The idea of DCA is simple: at each iteration, one replaces the second component h in the primal DC problem (P_{dc}) by its affine minorization, $h(x^k)+(x-x^k)^T y^k$, to generate the convex program

$$\min\{g(x)-(x-x^k)^T y^k, x\in R^n, y^k\in\partial h(x^k)\} \quad (10)$$

whose the solution set is $\partial g^*(y^k)$. Likewise, the second DC component g^* of the dual DC program (D_{dc}) is replaced by its affine minorization, $g^*(y^k)+(y-y^k)^T x^{k+1}$, to obtain a convex program whose the solution set is $\partial h(x^k)$.

In practice, a simplified form of the DCA is used. Two sequences $\{x^k\}$ and $\{y^k\}$ satisfying $y^k\in\partial h(x^k)$ are constructed, and x^{k+1} is a solution to the convex program (10). The simplified DCA scheme is described as follows.

```
Initialization: Choose an initial point $x^0 \in R^n$ and Let $k = 0$
Repeat
    Calculate $y^k \in \partial h(x^k)$
    Solve convex program (10) to obtain $x^{k+1}$
    Let k:=k+1
Until some stopping criterion is satisfied.
```

DCA is a descent algorithm without linesearch. The following properties are used in the next sections :(for simplicity, we omit the dual part of these properties).

- If the optimal value of problem (P_{dc}) is finite and the infinite sequence $\{x^k\}$ is bounded, then every limit point x^* of the sequence $\{x^k\}$ is a critical point of (P_{dc}).
- DCA converges linearly for general DC programs.
- If the second DC component h in (P_{dc}) is differentiable , then the subdifferential of the h at point x^k is reduced to a singleton, $\partial h(x^k)=\{\bigtriangledown h(x^k)\}$. In this case, x^{k+1} is a solution to the following convex program:

$$\min\{g(x)-(h(x^k)+\nabla h(x^k)^T(x-x^k)), x\in R^n\} \quad (11)$$

DCA is an efficient and robust algorithm for solving nonconvex problems, especially in the large-scale setting, and has been successfully applied to many nonconvex optimizations.

3. DC programming formulations for feature selection

Assume that the data for each class can be summarized by their moments, the mean and covariance. The problem of feature selection, given the moments, is approached in a worst case setting. Bhattacharyya proposed a SOCP framework for feature selection[4] based on the L_1-norm (referred to as the L_1-MPM):

$$\begin{aligned} \min_{w,b} \quad & \|w\|_1 && (12)\\ \text{s.t.} \quad & P\{X_1\in H_1\}\geqslant\delta && (13)\\ & P\{(X_2\in H_2\}\geqslant\delta && (14)\\ & X_1\sim(\mu_1,\Sigma_1), X_2\sim(\mu_2,\Sigma_2) && (15)\end{aligned}$$

where $\delta\in(0,1)$ is defined by the user. This optimization problem can be posed as a SOCP by using a multivariate generalization of the Chebychev-Cantelli inequality[14]. As mentioned before, feature selection via the L_1-norm is not a good idea since sparsity is an explicit goal of feature selection, while the usefulness of the minimizing of the L_0-norm is to yield a sparse classifier. Thus, in this work, we propose a feature selection framework based on the L_0-norm in which the data are summarized by their moments.

3.1. Linear version

In linear case, the aim of our work is to produce a linear classifier $f(x)=sgn(w^T x+b)$ with a low misclassification probability using a small set of useful features. This amounts to finding a sparse vector w.

3.1.1. Problem Definition

Replacing $\|w\|_1$ with $\|w\|_0$ in Eq.(12-15) yields an interesting formulation, called the L_0-MPM for short:

$$\begin{aligned} \min_{w,b} \quad & \|w\|_0 && (16)\\ \text{s.t.} \quad & P\{X_1\in H_1\}\geqslant\delta && (17)\\ & P\{(X_2\in H_2\}\geqslant\delta && (18)\\ & X_1\sim(\mu_1,\Sigma_1), X_2\sim(\mu_2,\Sigma_2) && (19)\end{aligned}$$

where parameter $\delta \in (0,1)$ is the lower bound on the classification accuracy in the worst-case setting. This optimization leads to a sparse classifier with a lower bound δ when data are summarized by their moments.

Specifically, minimizing L_0-norm in the objective function implements feature selection of the classifier while the two constraints state that the probability of a random vector taking values in a given half space is lower-bounded by δ. In other words, the probability of a random vector X_1 (or class X_2) taking values in the half space $H_1(w,b) = \{x|w^T x > b\}$ (or $H_2(w,b) = \{x|w^T x < b\}$) should be at least greater than the user defined parameter δ. The higher the value of parameter δ is , the more stringent is the requirement that all points belong to the correct half space.

The following should be noted:

- MPM mainly focused on maximizing the probability of predicting future data. Compared with the MPM, the main benefits of this L_0-MPM are that it yields a sparse classifier and thus can effectively control the dimensionality of the input data.
- Similar to the MPM, the L_0-MPM makes no specific distribution assumptions about the data distribution. Thus it is convenient to use in practical applications.
- Similar to the L_1-norm MPM, this formulation involves an explicit upper bound on the worst-case misclassification accuracy after selecting a subset of features. That is, we can control the tradeoff between the number of features and misclassification probability by controlling the bound δ.
- The L_0-MPM outperforms the existing the L_1-MPM in terms of a better ability of feature selection ability.

3.1.2. *Solving L_0-MPM*

The following multivariate generalization of the Chebyshev-Cantelli inequality [14] is subsequently used to derive an upper bound on the misclassification probability of a random vector taking values in a given half space.

Lemma 1. *Let X be a n dimensional random vector. The mean and covariance of X are $\mu \in R^n$ and $\Sigma \in R^{n\times n}$ respectively. Let $H(w,b) = \{z|w^T z < b, w \in R^n, w \neq 0, b \in R\}$ be a given half space. Then the following inequality holds:*

$$P\{X \in H\} \geqslant \frac{(b - w^T\mu)_+^2}{(b - w^T\mu)_+^2 + w^T\Sigma w} \tag{20}$$

where $(x)_+ = max\{x,0\}$.

Applying Lemma 1, the constraint for class X_1 in Eq.(16-19) can be handled by setting

$$P\{X_1 \in H_1\} \geqslant \frac{(w^T\mu_1 - b)_+^2}{(w^T\mu_1 - b)_+^2 + w^T\Sigma_1 W} \geqslant \delta \tag{21}$$

which results in two constraints:

$$w^T\mu_1 - b \geqslant \sqrt{\frac{\delta}{1-\delta}}\sqrt{w^T\Sigma_1 w} \tag{22}$$

$$w^T\mu_1 - b \geqslant 1 \tag{23}$$

For simplicity, we assume that both Σ_1 and Σ_2 are positive definite. Our results can be extended to general positive semi-definite cases. Then, let $\Sigma_1 = C_1C_1^T$ and $\Sigma_2 = C_2C_2^T, C_1, C_2 \in R^{n\times n}$. Similarly, by applying Eq.(20) to the other constraint, Eq.(16-19) can be formulated as

$$\min_{w,b} \; \|w\|_0 \tag{24}$$

$$\text{s.t. } w^T\mu_1 - b \geqslant \sqrt{\frac{\delta}{1-\delta}}\|C_1^T w\| \tag{25}$$

$$b - w^T\mu_2 \geqslant \sqrt{\frac{\delta}{1-\delta}}\|C_2^T w\| \tag{26}$$

$$w^T\mu_1 - b \geqslant 1, b - w^T\mu_2 \geqslant 1 \tag{27}$$

$$X_1 \sim (\mu_1, \Sigma_1), X_2 \sim (\mu_2, \Sigma_2) \tag{28}$$

with fixed $\delta \in (0,1)$. A a good approximation [11] of the $\|w\|_0$ would be

$$\|w\|_0 \approx \sum_{i=1}^{n} \eta(w_i) \tag{29}$$

where η is the function (see Fig.1) defined by

$$\eta(z) = 1 - \varepsilon^{-\alpha|z|}, \alpha \in R, \alpha > 0, \forall z \in R \tag{30}$$

Thus, the L_0-norm $\|w\|_0$ is approximated by:

$$\|w\|_0 \approx e^T(e - \varepsilon^{-\alpha|w|}), \alpha > 0 \tag{31}$$

With this approximation, the resulting optimization Eq.(24-28) takes the form:

$$\min_{w,b} \sum_{i=1}^{n} \eta(w_i) \tag{32}$$

$$\text{s.t. } w^T\mu_1 - b \geqslant \sqrt{\frac{\delta}{1-\delta}}\|C_1^T w\| \tag{33}$$

$$b - w^T\mu_2 \geqslant \sqrt{\frac{\delta}{1-\delta}}\|C_2^T w\| \tag{34}$$

$$w^T\mu_1 - b \geqslant 1, b - w^T\mu_2 \geqslant 1 \tag{35}$$

$$X_1 \sim (\mu_1, \Sigma_1), X_2 \sim (\mu_2, \Sigma_2) \tag{36}$$

Assuming that Ω is the feasible set of Eq.(32-36) and $\chi_\Omega(x)$ denotes the indicator function for the convex set Ω: $\chi_\Omega(x) = 0$ if $x \in \Omega, and +\infty$ otherwise. Let $x = (w,b) \in R^{n+1}$, and

$$g(x) = \alpha e^T|x| \tag{37}$$

$$h(x) = \alpha e^T|x| - e^T e + e^T \varepsilon^{-\alpha|x|}. \tag{38}$$

Obviously, $g(x)$ and $h(x)$ are both convex functions, and $\eta(x) = g(x) - h(x)$. Therefore, Eq.(32-36) can be reformulated as the following DC program:

$$\min\{g(x) + \chi_\Omega(x) - h(x)\} \tag{39}$$

Furthermore, it can be seen that $h(x)$ is differentiable everywhere and that $\nabla h(x) = (v, 0)$ with

$$v_j = \begin{cases} \alpha(1-\varepsilon^{-\alpha w_j}), & w_j \geqslant 0 \\ -\alpha(1-\varepsilon^{\alpha w_j}), & w_j < 0 \end{cases} \tag{40}$$

According to the generic DCA scheme, we solve the following convex program to obtain x^{k+1} in the k-th each iteration

$$\min\{g(x) + \chi_\Omega(x) - \nabla h(x^k)^T x\} \tag{41}$$

Let $|w| \leqslant t$. Solving the Eq.(41) amounts to solving the following SOCP for a fixed $\delta \in (0,1)$ with $\alpha > 0$

$$\min_{w,b,t} \alpha e^T t - (v^k)^T w \tag{42}$$

$$\text{s.t. } w^T\mu_1 - b \geqslant \sqrt{\frac{\delta}{1-\delta}}\|C_1^T w\| \tag{43}$$

$$b - w^T\mu_2 \geqslant \sqrt{\frac{\delta}{1-\delta}}\|C_2^T w\| \tag{44}$$

$$w^T\mu_1 - b \geqslant 1, b - w^T\mu_2 \geqslant 1 \tag{45}$$

$$-t \leqslant w \leqslant t \tag{46}$$

with $w, t \in R^n$ and $b \in R$. The problem can be effectively solved in polynomial time using the interior algorithm . Next we describe our DCA applied to Eq.(39).

Algorithm 1

Step 1. For a fixed $\delta \in (0,1), \varepsilon > 0$ is sufficiently small, and set k=0. Choose an initial point $x^0 \in \Omega$.

Step 2. Compute $\nabla h(x^k)$ via (40).

Step 3. Solve the SOCP (42-46) to obtain x^{k+1} .

Step 4. If either$\|x^{k+1} - x^k\| < \varepsilon$ or $g(x^{k+1}) - h(x^{k+1}) \geqslant g(x^k) - h(x^k) - \varepsilon$, stop and x^{k+1} is the computed solution. Otherwise, set k=k+1 and go to Step 2.

Theorem 1

- *Algorithm 1 generates a sequence $\{x^k\}$ such that $g(x^k) - h(x^k)$ decreases monotonously.*
- *The sequence $\{x^k\}$ converges linearly.*

Proof: These conclusion are direct consequences of the convergence properties of general DC program. The proof is then complete.

3.2. Nonlinear version

The approach in this paper can also be extended to formulate a nonlinear version using very few support vectors. Assume that the discriminating hyperplane is $\{x|\beta^T k(x) = b\}$ which divides the feature space into two subsets $\{x|\beta^T k(x) < b\}$ and $\{x|\beta^T k(x) > b\}$, where kernel k is a function obeying the Mercer conditions [8]. The $k(x)$ is a vector whose ith component is $k(x, x_i)$. In the feature space, we would like to find a sparse decision hyperplane

$$H(\beta, b) = \{x|\beta^T k(x) = b\} \tag{47}$$

with as few feature vectors as possible. According to the above analysis, this can be designed to minimize the cardinality of the set $S = \{i|\beta_i \neq 0\}$; in other words, this can be designed to minimize the L_0-norm of β.

Assume that $k_1 = k(X_1)$ is a random vector corresponding to class X_1 while $k_2 = k(X_2)$ is another random vector belonging to class X_2. Let the the means of k_1 and k_2 be $\overline{k_1}$ and $\overline{k_2}$ respectively, and the variances be $\overline{\Sigma_1}$ and $\overline{\Sigma_2}$ respectively. Using the Chebyshev bound, feature selection for the nonlinear MPM can be formulated as:

$$\min_{\beta,b} \|\beta\|_0 \tag{48}$$

$$\text{s.t. } \beta^T\overline{k_1} - b \geqslant \sqrt{\frac{\delta}{1-\delta}}\sqrt{\beta^T\overline{\Sigma_1}\beta} \tag{49}$$

$$b - \beta^T\overline{k_2} \geqslant \sqrt{\frac{\delta}{1-\delta}}\sqrt{\beta^T\overline{\Sigma_2}\beta} \tag{50}$$

$$\beta^T\overline{k_1} - b \geqslant 1, b - \beta^T\overline{k_2} \geqslant 1 \tag{51}$$

$$k_1 \sim (\overline{k_1}, \overline{\Sigma_1}), k_2 \sim (\overline{k_2}, \overline{\Sigma_2}) \tag{52}$$

Similarly, using the same approximation of the L_0-norm as in the linear case, we have

$$\|\beta\|_0 \approx \sum_{i=1}^{n} \eta(\beta_i) = e^T(e - \varepsilon^{-\alpha|\beta|}), \alpha > 0 \tag{53}$$

which leads to the following nonconvex optimization

$$\min_{\beta,b} \sum_{i=1}^{n} \eta(\beta_i) \tag{54}$$

$$\text{s.t. } \beta^T\overline{k_1} - b \geqslant \sqrt{\frac{\delta}{1-\delta}}\sqrt{\beta^T\overline{\Sigma_1}\beta} \tag{55}$$

$$b - \beta^T\overline{k_2} \geqslant \sqrt{\frac{\delta}{1-\delta}}\sqrt{\beta^T\overline{\Sigma_2}\beta} \tag{56}$$

$$\beta^T\overline{k_1} - b \geqslant 1, b - \beta^T\overline{k_2} \geqslant 1 \tag{57}$$

$$k_1 \sim (\overline{k_1}, \overline{\Sigma_1}), k_2 \sim (\overline{k_2}, \overline{\Sigma_2}) \tag{58}$$

For a fixed $\delta \in (0,1)$, this can also be reformulated as a DC program. Moreover, the solution of the Eq.(48-52) can be reached by solving Eq.(54-58). A corresponding DCA to solve this problem is constructed similarly. In this paper, we report the experimental results for this problem only.

4. Numerical experiments

To evaluate the proposed framework, numerical experiments are carried out on both a synthetic dataset and the real world data sets, Sonar, Pima, Ionosphere, Spam, WOBC, WPBC, WDBC and Madelon, from the University of California Irvine (UCI) Machine Learning Repository. The specifications of these datasets are summarized in Table 1.

Table 1. The description of datasets

datasets	instances	Features
Sonar	208	60
Pima	568	8
Ionosphere	350	34
Spam	200	57
WOBC	699	9
WPBC	198	32
WDBC	569	30
Madelon	200	500
Synthetic	15	1000

All of the methods were implemented in Matlab 7.0, and the experiments used the popular package SeDuMi[15] as a solver. In addition, 10-fold cross-validation was used in our experiments. The following performance criteria were used to evaluate the considered methods:

- Test-set accuracy (TSA), which is the classification correctness rate of all samples from two classes;
- Matthew's correlation coefficient (MCC)[16][17], which is a comprehensive measure of the quality of the classification model; the higher the value of the MCC is, the better the model is; The above values can be obtained from the decision function and are defined as[16]

$$TSA = \frac{TP+TN}{TP+FN+TN+FP} \tag{59}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{60}$$

  where TP and TN denote true positives and true negatives; FN and FP denote false negatives and false positives, respectively.
- Average number and percentage of selected features (PSF);
- Average number of iterations for the two iteration algorithms L_0-MPM and S-MPM[2].

The proposed method makes use of two parameters, α and δ. The Gaussian parameter α implements a tradeoff between accuracy and the number of features. These parameters should be optimized beforehand.

- The feature selection ability of the proposed method also depends on the choice of the bound parameter δ. From Table 2 and Fig.2, which illustrate the relationship between parameters α,δ and PSF in a linear setting,we observe that the number of selected features generally decreases as the value of α increases. For small values of δ , fewer features are reported, while as δ increases, the L_0-MPM selects a greater number of features.

- The accuracy of the proposed method depends heavily on the values of parameters α and δ. In a linear setting, the relationship between the TSA and the α is illustrated in Fig.3, while the relationship between the TSA and δ is illustrated in Fig.4.

Table 2. Feature numbers as α and δ for Spam dataset

$\alpha\backslash\ \delta$	0.9	0.7	0.5	0.3	0.1
50	3.7	3.4	2.9	2.3	2.0
10	3.7	3.4	3.0	2.4	2.0
5	3.7	3.4	3.1	2.5	2.0
1	3.9	3.4	3.3	2.5	2.0
0.5	3.9	3.4	3.3	2.5	2.0
0.1	3.9	3.5	3.4	2.5	2.0

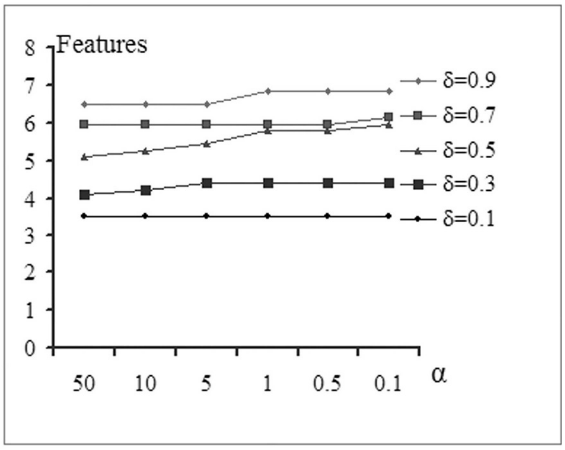

Fig.2 PSF versus α for various values of δ for Spam data

We find that the accuracy of the L_0-MPM increases when δ ranges from 0.1 to 0.9, and that the L_0-MPM produces greater accuracy when α is set to a larger value. These findings were helpful in the choice of parameters α and δ in the following benchmark experiments.

According to the above analysis, parameters $\alpha = 5$ and $\delta = 0.9$ were chosen for our experiments on the UCI datasets, except the Ionosphere dataset, for which we set $\alpha = 0.5$.

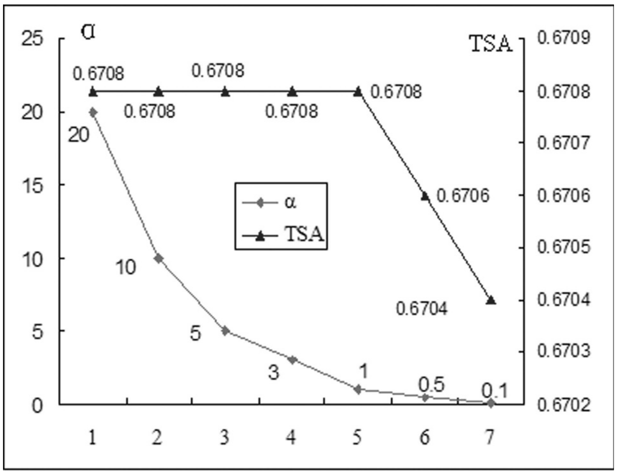

Fig.3 TSA versus parameter α for Spam data set

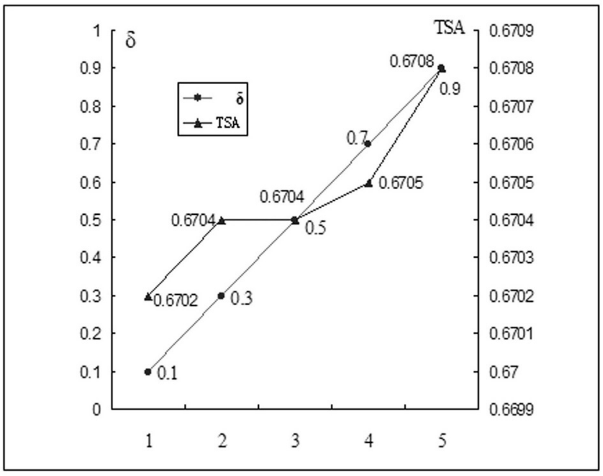

Fig.4 TSA versus the lower bound δ for Spam data set

4.1. *Experiments on UCI datasets*

Here we considered two experiments. First, we compared our linear L_0-MPM with the linear MPM,

L_1-MPM and S-MPM. Then, we compared our nonlinear L_0-MPM with the nonlinear MPM and L_1-MPM.

4.1.1. Experiments on linear versions on UCI datasets

We compared the linear L_0-MPM with the linear MPM, L_1-MPM and S-MPM on eight UCI datasets. The TSA and MCC for the three methods,L_0-MPM, MPM and L_1-MPM, in a linear setting, are summarized in Table 3. In addition, a comparison of the PSF for the three methods is presented in Fig.5.

Table 3. Comparison of L_0-MPM, MPM and L_1-MPM in terms of TSA and MCC

dataset	criteria	MPM (%)	L_1-MPM (%)	L_0-MPM (%)
Sonar	TSA	75.10	77.89	79.50
	MCC	61.71	68.06	70.89
Pima	TSA	73.80	65.53	65.79
	MCC	45.46	42.86	44.91
Ionosphere	TSA	85.40	71.18	85.74
	MCC	79.10	53.51	82.58
Spam	TSA	67.02	60.57	67.08
	MCC	55.36	23.89	62.41
WOBC	TSA	91.61	89.46	75.26
	MCC	80.05	78.00	72.56
WPBC	TSA	24.21	62.11	61.58
	MCC	13.06	25.03	24.99
WDBC	TSA	90.71	90.89	88.75
	MCC	82.42	83.65	85.63
Madelon	TSA	60.30	60.58	61.30
	MCC	23.81	22.89	31.98

Comparison of the L_0-MPM and MPM.
The results from Table 3 show that the proposed L_0-MPM does not outperform the other two methods on all datasets. In terms of the TSA criterion, the performance of the L_0-MPM is not significantly different to that of the MPM on four datasets (Ionosphere, Spam, WDBC, and Madelon). On the WPBC and Sonar datasets, the L_0-MPM is superior to the MPM with respect to the TSA criterion, whereas the MPM performs slightly better than our method on the WOBC and Pima datasets. In addition, with respect to the MCC, we find that the L_0-MPM is superior to the MPM in six of the eight datasets. Meanwhile, Fig.5 shows that our model can always suppress many more features than the MPM. These results show that our model does not have an adverse effect on generalization, yet always selects fewer features than the MPM.

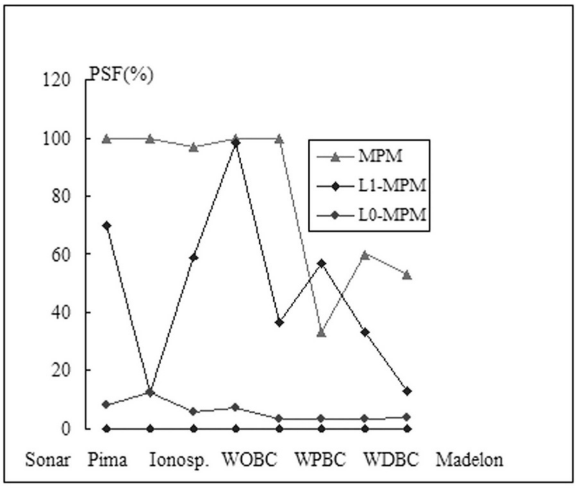

Fig.5 Three methods for PSF in UCI datasets

Comparison of the L_0-MPM and L_1-MPM.
Table 3 shows that the L_0-norm MPM shows no significant difference compared with the L_0-MPM with respect to TSA analysis in six of the eight datasets. However, Fig. 5 shows that the L_0-MPM reduces the number of features considerably with the percentage of suppressed features by our method varying from 3 to 20%. Moreover, the L_0-MPM consistently realizes better performance than the L_1-MPM with respect to PSF analysis in seven datasets, excluding Pima, while the two models have the same PSF results for Pima. Moreover, in terms of MCC, we find that the linear L_0-MPM is superior to the L_1-MPM in six of the eight datasets, while in the other two datasets the performances of the two methods show no significant difference. These results suggest that, without loss of generalization, the proposed linear framework always selects fewer features than the linear L_1-MPM.

To further evaluate the performances of the three algorithms, average ranks for these are given in Table 4, from which we observe that the average rank of the L_0-MPM is lower than that of the MPM and L_1-MPM. Moreover, Fig.5 show that our L_0-MPM suppress much more features than the L_1-MPM and

MPM. A possible reason for these is that the L_1-MPM improves classification accuracy by removing irrelevant features compared with the MPM. In addition, the L_0-MPM, without loss of generalization, suppresses many more features than the L_1-MPM by minimizing the L_0-norm instead of the L_1-norm as is the case in the latter method.

Comparison of the L_0-MPM and S-MPM.

In this section, we present a comparison of the L_0-MPM and S-MPM with respect to TSA, MCC, PSF, and number of iterations. The average results on the three datasets, Ionosphere, Sonar, and Spam, are given in Tables 5, 6, and 7, respectively.

According to the above analysis, we find that the PSF for the L_0-MPM is noticeably lower than that for the L_1-MPM and MPM in most cases. This means that our method does not have an adverse effect on the generalization of the MPM by removing many irrelevant features, while the performance in feature selection is better than that of the L_1-MPM. Thus, one notable benefit of the proposed method is its efficiency in feature selection.

For the TSA comparison, Table 5 reports that the performances of the two models are similar on all three datasets. At the same time, Tables 6 and 7 show that the performance of the L_0-MPM is quite good compared with that of the S-MPM according to PSF and the number of iterations on all three datasets. In addition, Table 6 shows that the PSF of the L_0-MPM is noticeably lower than that of the S-MPM on two of the three datasets, which shows that the L_0-MPM outperforms the S-MPM in feature selection.

Table 4. Comparison with MPM and L_1-MPM for ranks

datasets	MPM	L_1-MPM	L_0-MPM
Sonar	3	2	1
Pima	1	2	3
Ionosphere	2	3	1
Spam	2	3	1
WOBC	1	2	3
WPBC	3	1	2
WDBC	2	1	3
Madelon	3	2	1
Synthetic data	2.5	1	2.5
Average rank	1.83	2.0	1.72

Table 5. Comparison of TSA and MCC between L_0-MPM and S-MPM

dataset	criteria	Ionosp (%)	Spam (%)	Sonar (%)
S-MPM	TSA	85.12	79.00	78.65
	MCC	78.02	60.20	68.06
L_0-MPM	TSA	85.74	67.08	79.50
	MCC	82.58	62.41	70.89

Table 6. Comparison of PSF (%) between L_0-MPM and S-MPM

dataset	Ionosp	Spam	Sonar
S-MPM	14.71	21.05	8.33
L_0-MPM	5.88	7.02	8.33

Table 7. Comparison of the number of iterations between L_0-MPM and S-MPM

dataset	Ionosp	Spam	Sonar
S-MPM	5	8	34
L_0-MPM	2	2	3

4.1.2. Experiments on the nonlinear versions on UCI datasets

For the nonlinear case, we used the popular Gaussian kernel with the kernel width σ selected over the range $\{2^i | i = -3, -2, \cdots, 3\}$ for each dataset. We compared the nonlinear L_0-MPM with the nonlinear L_1-MPM and MPM. Investigation of the TSA and PSF was carried out using four datasets: Ionosphere, Sonar, WDBC, and Madelon. The results are illustrated in Figs. 6 and 7, from which we find that the nonlinear L_0-MPM is superior to the nonlinear L_0-MPM and MPM in both generalization and feature selection ability on all four datasets.

4.2. Experiments on a synthetic dataset

Consider a synthetic dataset generated as follows. The class label, y, of each observation was randomly chosen to be 1 or -1 with probability 0.5. The first ten features of the observation, x, were drawn as$yN(-i,1)$, where $N(\mu,\sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. In total, 990 other features were drawn as $N(0,1)$ with 50 such observations generated. The feature selection problem was to detect the first 10 features since these are the most discriminatory of the given 1000 features. The generation of the synthetic data is reported in Table 1.

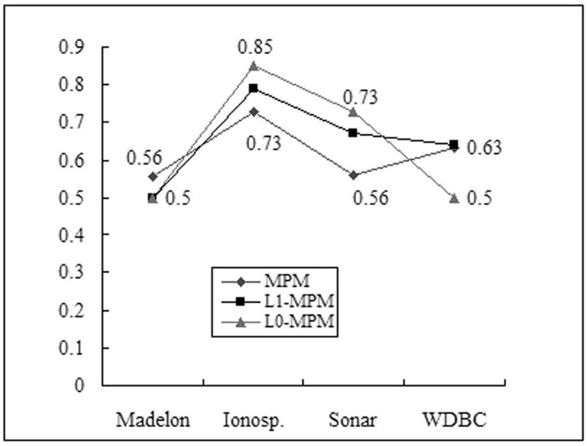

Fig. 6 Three methods for TSA in nonlinear setting

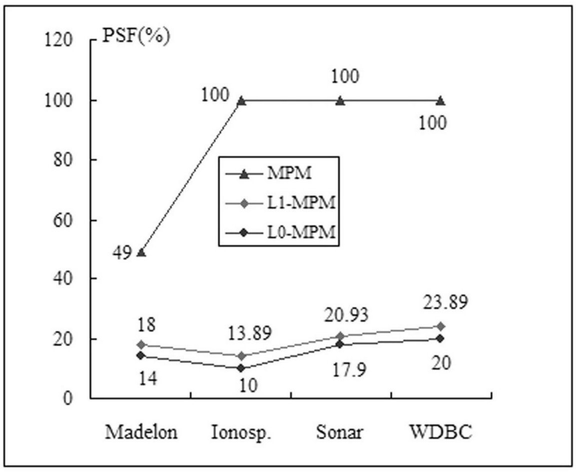

Fig. 7 Three methods for PSF in nonlinear setting

We also experimented with the proposed L_0-MPM using different values of α on the synthetic dataset. The experiment results are depicted in Fig. 8, from which we find that the L_0-MPM reports greater accuracy as the value of α increases. Finally, we set α=10 for the synthetic dataset. Moreover, we find that as the value of δ increases the formulation reports more discriminatory features.

Our L_0-MPM provides a good classification: the correctness of the classification on the test set varies from 46.78 to 86.67%, while the percentage of suppressed features varies from 55 to 90%. For α=10 and δ =0.9 in the 10 repeated experiments, the corresponding list of the number of features selected by the L_0-MPM and L_1-MPM are $\{9,7,11,12,8,11,12,10,11,10\}$ and $\{9,7,11,12,8,11,12,10,11,10\}$, respectively. The numbers of features selected by the MPM are $\{937,930,926,932,939,935,930,933,932,936\}$ for the 10 repeated experiments. These results show that despite sample size being low compared with the number of features, the L_0-MPM formulation is able to discover the most discriminatory features.

In addition, we also present a map to illustrate the comparison of our linear version with the linear L_1-MPM and MPM. Experimental results averaged over the 10 repeated experiments are presented in Fig. 9.

From Figs.8 and 9 we see that the L_0-MPM is competitive with the L_1-MPM and MPM. However, our L_0-MPM considerably reduces the number of features compared with the L_1-MPM and MPM. The three formulations identify most of the discriminatory features, but the L_0-MPM formulation selects more features. This demonstrates that the L_0-MPM identifies discriminatory features and is comparable with the L_1-MPM and MPM in terms of generalization.

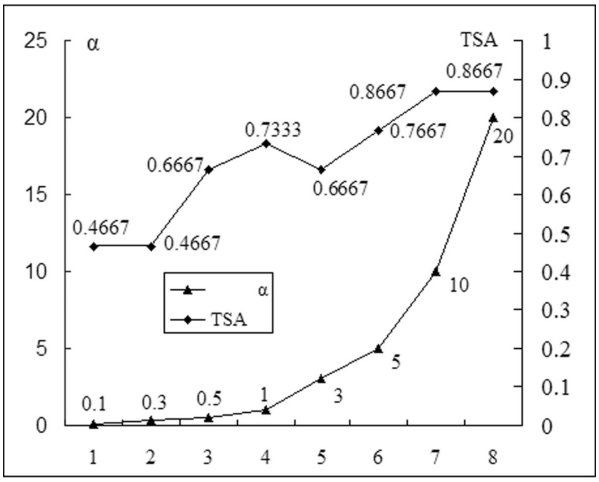

Fig.8 TSA versus parameter α for Synthetic data

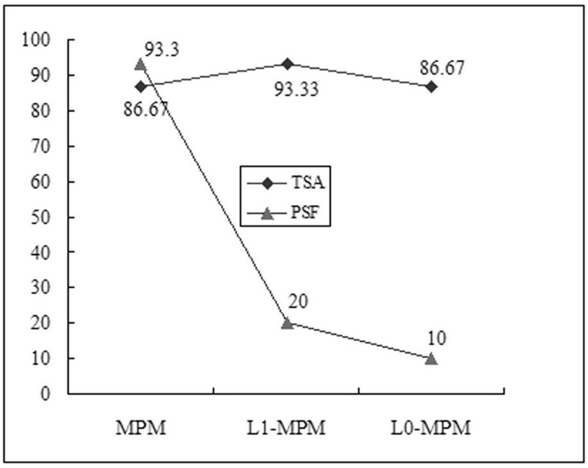

Fig.9 Three methods in TSA and PSF for Synthetic data

From the above analysis, we find that in terms of generalization, the L_0-MPM is almost the same as the L_1-MPM and S-MPM, but contains fewer features than these existing methods.

5. Conclusions and future directions

We constructed a new feature selection framework based on the L_0-norm in which the data are summarized by their moments. By applying suitable DC decomposition to the L_0-norm, we presented several DC program formulations for the proposed framework using the moments of the dataset. Moreover, problems can be solved effectively using the DCA. The resulting DCA needs to solve only successive SOCPs and converges linearly.

The feature selection abilities of the proposed formulations were tested on both synthetic and real world datasets. Experiments show that, compared with existing methods, the proposed framework either improves or shows no significant difference in generalization, yet suppresses more features. A possible reason for this is that removing irrelevant features in classifications does not reduce the generalization of classifiers.

- Compared with the original MPM, the proposed L_0-MPM improves accuracy by removing irrelevant features in almost all cases.
- Compared with other feature selection approaches, L_1-MPM and S-MPM, this L_0-MPM does not sacrifice generalization in selecting fewer features.

The above results suggest that applying the L_0-norm to feature selection yields a particular benefit: it is an ideal approach for enforcing sparsity without sacrificing classification performance. However, finding more effective approximations of the L_0-norm would be interesting. Better methods to optimize the L_0-norm directly will be investigated in future work.

The construction of DCA involves DC components g and h but not the function f itself. Hence, for a DC program, each DC decomposition corresponds to a different version of DCA. Currently, the question of ”*good*” DC decompositions of L_0-MPM is still open, and works in these directions are in progress.

In this paper, we have only considered the binary cases because multi-class problems can be easily approached via standard techniques, such as the one vs. others and the one vs. one technique.

Acknowledgments

This work is supported by National Nature Science Foundation of China (11271367) and the Research Projects of Teachers (2010JS052).

References

1. Z. Wei and D.Miao,”N-grams based feature selection and text representation for Chinese Text Classification”, *Int. J. Comput. Intell.Syst.* , **2(4)**, 365-374 (2009).
2. L.M.Yang,L.S.Wang, Y.H. Sun and R.Y. Zhang, ”Simultaneous feature selection and classification via Minimax Probability Machine”, *Int. J. Comput. Intell.Syst.* ,**3(6)**, 754-760 (2010).
3. ZH.H. Deng, SH.T.Wang, F.L.Chung. A minimax probabilistic approach to feature transformation for multi-class data. *Appl soft comput*, **13** ,116-127(2013).
4. C. Bhattacharyya, ”Second order cone programming formulations for feature selection”, *J.Mach. Learn.Res.*, **5**,1417-1433(2004).
5. M. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, ”Applications of second order cone programming”, *Linear Algebra Appl.*, **284**, 193-228 (1998).

6. G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan, ”Minimax probability machine”, *Adv. Neural. Inf. Process.*, **14**(2002).
7. K. Yoshiyama, A. Manifold-regularized minimax probability machine. *In: Partially Supervised Learning, First IAPR TC3 Workshop* ,**7018**, pp. 42-51 (2012).
8. V.N.Vapnik, ” Statistical Learning Theory”, New York, Wiley.1998.
9. P.D.Tao, L.T.H An, ” Convex analysis approaches to DC programming: theory, algorithms and applications”, *Acta. Math*, **22(1)**, 287-367(1997).
10. H.A.Le Thi, T.P. Dinh,” The DC Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems”, *Ann. Oper. Res* , **133**, 23-46(2005).
11. H.A.Le Thi ,H.M. Le, V.V.Nguyen,P.D.Tao, ”A DC programming approach for feature selection in support vector machines learning”, *Adv.Data Anal.Classif.*, **2** ,259-278(2008).
12. W.Guan,A.Gray. Sparse high-dimensional fractional-norm support vector machine via DCprogramming. *Comput Stat Data Anal.***67**,136-148(2013).
13. Y.Saeki,D.Kuroiwa. Optimality conditions for DC programming problems with reverse convex constraints. *Nonlinear Anal.* **80**, 18-27(2013).
14. W. Marshall and I. Olkin, ”Multivariate Chebychev inequalities”, *Annals of Math. Stat.*, **31(4)**,1001-1014(1960).
15. J.F. Sturm, ”Using SeDuMi 1.03, a MATLAB toolbox for optimization over symmetric cones”, (1999). *http://www.Unimaas.nl/sturm/software/sedumi.html.*
16. L.M. Yang, Q.Sun. Recognition of the hardness of licorice seeds using a semi-supervised learning method and near-infrared spectral data. *Chemom. Intell. Lab. Syst.* **114**, 109-115(2012).
17. H.Xu, Z.C.Liu, W.S.Cai, X.G. Shao. A wavelength selection method based on randomization test for near-infrared spectral analysis. *Chemom. Intell. Lab. Syst.*, **97**.189-193(2009).