Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 416320, 11 pages
http://dx.doi.org/10.1155/2013/416320

Research Article

A Semisupervised Feature Selection with Support Vector Machine

Kun Dai, Hong-Yi Yu, and Qing Li

National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China

Correspondence should be addressed to Kun Dai; daikun_1223@163.com

Received 15 May 2013; Accepted 4 October 2013

Academic Editor: Antonio J. M. Ferreira

Feature selection has proved to be a beneficial tool in learning problems with the main advantages of interpretation and generalization. Most existing feature selection methods do not achieve optimal classification performance, since they neglect the correlations among highly correlated features which all contribute to classification. In this paper, a novel semisupervised feature selection algorithm based on support vector machine (SVM) is proposed, termed SENFS. In order to solve SENFS, an efficient algorithm based on the alternating direction method of multipliers is then developed. One advantage of SENFS is that it encourages highly correlated features to be selected or removed together. Experimental results demonstrate the effectiveness of our feature selection method on simulation data and benchmark data sets.

1. Introduction

Feature selection, with the purpose of selecting relevant feature subsets among thousands of potentially irrelevant and redundant features, is a challenging topic of pattern recognition research that has attracted much attention over the last few years. A good feature selection method has several advantages for a learning algorithm such as reducing computational cost, increasing its classification accuracy, and improving result comprehensibility [1].

Considering the usage of the class label information, feature selection methods can be classified into supervised methods, unsupervised methods, and semisupervised methods. Supervised feature selection methods usually use only information from labeled data to find the relevant feature subsets [2–4]. However, in many real, world applications, the labeled data are very expensive or difficult to obtain, which brings difficulty to create a large training data set. This situation arises naturally in practice, where large amount of data can be collected automatically and cheaply, when manual labeling of samples remains difficult, expensive and time consuming. Unsupervised feature selection methods could be an alternative in this case through exploiting the information conveyed by the large amount of unlabeled data [5, 6]. However, as these unsupervised algorithms ignore label information, important hints from labeled data are left out and this will generally downgrade the performance of unsupervised feature selection algorithms. The combination of both supervised methods and unsupervised methods is semisupervised approaches [7–10] which exploit the information of both labeled and unlabeled data. A good survey about semisupervised feature selection approaches can be found in [9].

The performances of the most existing semisupervised feature selection methods are insufficient when there are several highly correlated features, which are all relevant to classification and the way they interact can help with the interpretability of the objective problem [11]. Given these premises, this paper provides two main contributions as follows.

(i) We present a novel semisupervised feature selection scheme based on support vector machine (SVM) and the elastic net penalty proposed by Zou and Hastie [12] combining l_1 and l_2 regularizations, termed SENFS.

(ii) In order to solve SENFS with the nondifferentiability of both the loss function and the l_1-norm regularization term, an efficient algorithm based on the

alternating direction method of multipliers (ADMM) [13] is developed.

Compared with other semisupervised feature selection algorithms, SENFS provides the following benefits.

(i) It permits highly correlated features to be selected or removed together.

(ii) It performs automatic feature selection as part of the training process and can achieve better classification performance using the selected features.

The effectiveness of SENFS is validated on simulated data and six benchmark semisupervised data sets. Our main finding is that SENFS can identify the features that are relevant to classification using the data set that consists of only a few labeled samples and many unlabeled samples.

This paper is organized as follows. Section 2 briefly introduces the methodology. In Section 3, we derive an iterative algorithm that yields the entire solution path based on ADMM to solve this proposed method. In Section 4, we evaluate the performance of this proposed method on both simulated and real-world data, followed by a summary in Section 5.

2. Methodology

Assume that all samples sampled from the same population generated by target concept consist of p features. Given a set of samples $\mathbf{X} = (x_1, \ldots, x_n)^T$, in which n is the number of samples, the ith sample or input vector x_i of original feature D with p features is denoted by $x_i = \{x_{i1}, x_{i2}, \ldots, x_{ip}\}$. The set $\mathbf{X}$ can be divided into two parts: labeled set $\mathbf{X}_l = (x_1, \ldots, x_l)^T$ for which labels $\mathbf{y}_l = (y_1, \ldots, y_l)^T$ are provided with $y_i \in \{-1, 1\}_{i=1}^l$ for binary problem and unlabeled set $\mathbf{X}_u = (x_{l+1}, \ldots, x_{l+u})^T$ whose labels are not given, where l and u are the number of labeled and unlabeled samples, respectively, and $n = l + u$. Then, the generic goal of semisupervised feature selection is to find a feature subset D' with d ($d < p$) features which contains the most informative features using both data information of $\mathbf{X}_l$ and $\mathbf{X}_u$. In other words, the samples $(x'_1, \ldots, x'_n)^T$ represented in the d-dimensional space can well preserve the information of the samples $\mathbf{X} = (x_1, \ldots, x_n)^T$ represented in the original p-dimensional space.

We begin our discussion with the binary supervised feature selection based on the elastic net penalty. Wang et al. [11] proposed a supervised feature selection method based on SVM with the elastic net penalty term named doubly regularized support vector machine for binary classification problems, which solves the optimization of the following generic objective function over both the hyper plane parameters (β, β_0):

$$\min_{\beta,\beta_0} C\sum_{i=1}^{n} V\left(y_i g\left(x_i\right)\right) + \lambda_1 \|\beta\|_1 + \frac{\lambda_2 \|\beta\|_2^2}{2}, \tag{1}$$

where the decision function is defined as $g(x) = x\beta + \beta_0$ and both λ_1 and λ_2 are tuning parameters, and C is the regularization parameter. V is the margin loss function; for example, hinge loss $V(z) = H_1(z) = \max(1 - z, 0)$. The role of the l_1-norm penalty is to allow selection, and the role of the l_2-norm penalty is to help groups of highly correlated features get selected or removed together which is denoted by the grouping effect [11].

As for semisupervised feature selection, considering $\mathbf{X}_l$ and $\mathbf{X}_u$, inspired by the semisupervised learning algorithm TSVM [14], we apply the elastic net penalty for semisupervised feature selection (SENFS), which solves the following optimization task over both the hyper plane parameters (β, β_0) and the unlabeled vector $\mathbf{y}_u = (y_{l+1}, \ldots, y_n)^T$:

$$\begin{aligned} \min_{\beta,\beta_0,\mathbf{y}_u \in \{-1,+1\}} \quad & C\sum_{i=1}^{l} V\left(y_i g\left(x_i\right)\right) \\ & + C^* \sum_{i=l+1}^{n} U\left(y_i g\left(x_i\right)\right) + \lambda_1 \|\beta\|_1 + \frac{\lambda_2 \|\beta\|_2^2}{2} \\ \text{s.t.} \quad & \frac{1}{u}\sum_{i=l+1}^{n} \max\left(0, y_i\right) = c. \end{aligned} \tag{2}$$

The constraint in (2) is called the balancing constraint and is necessary to avoid the trivial solutions where all unlabeled samples are assigned to the same class. This constraint enforces a manually chosen constant c and an approximation of this constraint writes $(1/n)\sum_{i=1}^{n} g(x_i) = 2c - 1$ [15] with c. So the constraint can be rewritten as $(1/u)\sum_{i=l+1}^{n} g(x_i) = (1/l)\sum_{i=1}^{l} y_i$. V and U employ the same loss; for example, hinge loss $V(z) = H_1(z) = \max(1 - z, 0)$.

Obviously, the difficulty of the above optimization task consists in finding the optimal assignment for the unlabeled vector y and the hyper plane parameters (β, β_0), which is a mixed-integer programming problem [16]. As described in [15], for a fixed (β, β_0), $\arg\min_y V(yg(x)) = \operatorname{sign}(g(x))$. So the problem of (2) can be seen equivalently as

$$\begin{aligned} \min_{\beta,\beta_0} \quad & C\sum_{i=1}^{l} V\left(y_i g\left(x_i\right)\right) \\ & + C^* \sum_{i=l+1}^{n} U\left(\left|g\left(x_i\right)\right|\right) + \lambda_1 \|\beta\|_1 + \frac{\lambda_2 \|\beta\|_2^2}{2} \\ \text{s.t.} \quad & \frac{1}{u}\sum_{i=l+1}^{n} g\left(x_i\right) = \frac{1}{l}\sum_{i=1}^{l} y_i. \end{aligned} \tag{3}$$

On the other hand, one effective approximation of the loss function $U(|z|)$ was a clipped variant [17] which can be expressed as

$$U\left(|z|\right) = R_s\left(z\right) + R_s\left(-z\right) - \left(1 - s\right), \tag{4}$$

where $R_s(z)$ is the Ramp loss defined as $R_s(z) = H_1(z) - H_s(z)$ with $H_s(z) = \max(s - z, 0)$, $0 \le s < 1$. In our experiments, the typical value of s is 0.3. The main reason to use the clipped

symmetric hinge loss is the gain of sparsity in the number of support vectors yielded by the optimizer [17]. Then we can get

$$\begin{aligned}
&\min_{\beta,\beta_0} C^* \sum_{i=l+1}^{n} U\left(\left|g\left(x_i\right)\right|\right) \\
&\quad = \min_{\beta,\beta_0} C^* \sum_{i=l+1}^{n} \left(R_s\left(z\right) + R_s\left(-z\right) - \left(1-s\right)\right) \\
&\quad = \min_{\beta,\beta_0} C^* \sum_{i=l+1}^{n} \left(H_1\left(g\left(x_i\right)\right) + H_1\left(-g\left(x_i\right)\right)\right) \\
&\qquad - C^* \sum_{i=l+1}^{n} \left(H_s\left(g\left(x_i\right)\right) + H_s\left(-g\left(x_i\right)\right)\right).
\end{aligned} \tag{5}$$

From (5), we can know that solving the optimization problem (3) with the clipped symmetric hinge loss is equivalent to solving a classical SVM with the unlabeled samples counted twice with $y_i = 1$ when $l+1 \le i \le l+u$ and $y_i = -1$ when $l+u+1 \le i \le l+2u$, which are artificial labels. Therefore problem (3) can be rewritten as

$$\begin{aligned}
\min_{\beta,\beta_0} \quad & C\sum_{i=1}^{l} V\left(y_i g\left(x_i\right)\right) \\
& + C^* \sum_{i=l+1}^{l+2u} \left(H_1\left(y_i g\left(x_i\right)\right) - H_s\left(y_i g\left(x_i\right)\right)\right) \\
& + \lambda_1 \|\beta\|_1 + \frac{\lambda_2 \|\beta\|_2^2}{2} \\
\text{s.t.} \quad & \frac{1}{u}\sum_{i=l+1}^{n} g\left(x_i\right) = \frac{1}{l}\sum_{i=1}^{l} y_i.
\end{aligned} \tag{6}$$

As we can seen from (6), when $C \neq 0$, $C^* = 0$, SENFS evolves into a supervised feature selection algorithm, and when $C = 0$, $C^* \neq 0$, it becomes an unsupervised model.

In the following, we will illustrate how SENFS has the grouping effect for correlated features. The following theorem describes this point.

Theorem 1. *Denote the solution to* (6) *by* $(\widetilde{\beta}, \widetilde{\beta}_0)$ *and* $\widehat{\beta}_0$*, the input* j*th feature by* F_j*, and the input* k*th feature by* F_k*. Then for any pair* (j,k)*, one can have*

$$\begin{aligned}
\left|\widetilde{\beta}_j - \widetilde{\beta}_k\right| \le & \frac{2M}{\lambda_2}\sum_{i=1}^{l}\left|x_{ij} - x_{ik}\right| \\
& + \frac{2M'}{\lambda_2}\sum_{i=l+1}^{l+2u}\left|x_{ij} - x_{ik}\right|,
\end{aligned} \tag{7}$$

where M *and* M' *are positive finite constants. Furthermore, if the input features* F_j *and* F_k *are centered and normalized, then*

$$\left|\widetilde{\beta}_j - \widetilde{\beta}_k\right| \le \frac{2}{\lambda_2}\left(\sqrt{2l\left(1-\rho_l\right)} + 2\sqrt{u\left(1-\rho_u\right)}\right), \tag{8}$$

where ρ_l *and* ρ_u *are the sample correlations between* F_j *and* F_k, $\rho_l = \sum_{i=1}^{l}(x_{ij} - \overline{x}_{\cdot j})(x_{ik} - \overline{x}_{\cdot k}) / \sqrt{\sum_{i=1}^{l}(x_{ij} - \overline{x}_{\cdot j})^2 \sum_{i=1}^{l}(x_{ik} - \overline{x}_{\cdot k})^2}$, $\overline{x}_{\cdot j} = (1/l)\sum_{i=1}^{l} x_{ij}$, $\overline{x}_{\cdot k} = (1/l)\sum_{i=1}^{l} x_{ik}$, *and* $\rho_k = \sum_{i=l+1}^{l+2u}(x_{ij} - \overline{x}'_{\cdot j})(x_{ik} - \overline{x}'_{\cdot k}) / \sqrt{\sum_{i=l+1}^{l+2u}(x_{ij} - \overline{x}'_{\cdot j})^2 \sum_{i=l+1}^{l+2u}(x_{ik} - \overline{x}'_{\cdot k})^2}$, $\overline{x}'_{\cdot j} = (1/2u)\sum_{i=l+1}^{l+2u} x_{ij}$, $\overline{x}'_{\cdot k} = (1/2u)\sum_{i=l+1}^{l+2u} x_{ik}$.

The term $|\widetilde{\beta}_j - \widetilde{\beta}_k|$ in (7) describes the difference between the coefficient paths of F_j and F_l. If both features are highly correlated, that is, $\rho = 1$, Theorem 1 says that difference between the coefficient paths of them is almost 0, in which case both features will be selected or removed together. The upper bound in (7) or (8) provides a quantitative description for the grouping effect of SENFS.

3. Algorithm for SENFS

The alternating direction method of multipliers (ADMM) developed in the 1970s and is well suited to distributed convex optimization and in particular to large-scale problems arising in statistics, machine learning, and related areas. The method is closely related to many other algorithms, such as the method of multipliers [18], Douglas-Rachford splitting [19], Bregman iterative algorithms [20] for l_1 problems, and others.

In this section, we first propose an efficient algorithm to solve SENFS based on ADMM by introducing auxiliary variables and reformulating the original problem. Then prove its convergence property and get the adjustment principle for penalty parameters. Finally describe the stopping criterion and computational cost.

3.1. Deriving ADMM for SENFS. It is hard to solve the model (6) directly due to the nondifferentiability of three loss functions and a l_1-norm term. In order to derive an ADMM algorithm, we introduce some auxiliary variables to handle these nondifferentiable terms.

Let $X_l = \{(x_{ij})_{i=1,j=1}^{l,p}\}$ denote labeled data, let $X_u = \{(x_{ij})_{i=l+1,j=1}^{l+2u,p}\}$ denote unlabeled data, and let Y_l, Y_u be diagonal matrixes with their diagonal elements to be the vector $\mathbf{y}_l = (y_1, \ldots, y_l)^T$ and $\mathbf{y}_u = (y_{l+1}, \ldots, y_{l+2u})^T$, respectively. The constrained problem in (6) can be reformulated into an equivalent form

$$\begin{aligned}
\min_{\beta,\beta_0} \quad & C\sum_{i=1}^{l}(h_i)_+ + C^*\sum_{i=l+1}^{l+2u}(a_i)_+ \\
& - C^*\sum_{i=l+1}^{l+2u}(b_i)_+ + \lambda_1\|\mathbf{t}\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2 \\
\text{s.t.} \quad & \begin{cases}
\mathbf{h} = \mathbf{1}_l - Y_l\left(X_l\beta + \beta_0\mathbf{1}_l\right) \\
\mathbf{a} = \mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) \\
\mathbf{b} = s\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) \\
\mathbf{t} = \beta \\
\dfrac{1}{u}\displaystyle\sum_{i=l+1}^{n}\left(x_i\beta + \beta_0\right) = r,
\end{cases}
\end{aligned} \tag{9}$$

where $\mathbf{h} = (h_1, \ldots, h_l)^T$, $\mathbf{a} = (a_1, \ldots, a_{2u})^T$, and $\mathbf{b} = (b_1, \ldots, b_{2u})^T$, $\mathbf{1}_l$ is an l-column vector of 1s and $\mathbf{1}_u$ is an $2u$-column vector of 1s, and $r = (1/l)\sum_{i=1}^{l} y_i$. The Lagrangian function of (9) is

$$\begin{aligned} l(\gamma) = {} & C\sum_{i=1}^{l}(h_i)_+ \\ & + C^*\sum_{i=l+1}^{l+2u}\left((a_i)_+ - (b_i)_+\right) + \lambda_1\|\mathbf{t}\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2 \\ & + \mathbf{u}_h^T\left(\mathbf{1}_l - Y_l\left(X_l\beta + \beta_0\mathbf{1}_l\right) - \mathbf{h}\right) \\ & + \mathbf{u}_a^T\left(\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) - \mathbf{a}\right) \\ & + \mathbf{u}_b^T\left(s\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) - \mathbf{b}\right) \\ & + \mathbf{v}^T(\beta - \mathbf{t}) + q\left(r - \frac{1}{u}\sum_{i=l+1}^{n}\left(x_i\beta + \beta_0\right)\right). \end{aligned} \tag{10}$$

In problem (10), $\gamma = \{\beta, \beta_0, \mathbf{h}, \mathbf{a}, \mathbf{b}, \mathbf{t}, \mathbf{u}_h, \mathbf{u}_a, \mathbf{u}_b, \mathbf{v}, q\}$, and $\mathbf{u}_h$, $\mathbf{u}_a$, and $\mathbf{u}_b$ are dual variables corresponding to the constraints $\mathbf{h} = \mathbf{1}_l - Y_l(X_l\beta + \beta_0\mathbf{1}_l)$, $\mathbf{a} = \mathbf{1}_u - Y_u(X_u\beta + \beta_0\mathbf{1}_u)$, and $\mathbf{b} = s\mathbf{1}_u - Y(X_u\beta + \beta_0\mathbf{1}_u)$, respectively, $\mathbf{v}$ is corresponding to the constraint $\mathbf{t} = \beta$, and q is a scalar corresponding to the balancing constrain. As in the method of multipliers, we form the augment Lagrangian

$$\begin{aligned} L(\gamma) = {} & l(\gamma) + \frac{\mu_1}{2}\left\|\mathbf{1}_l - Y_l\left(X_l\beta + \beta_0\mathbf{1}_l\right)\right\|_2^2 \\ & + \frac{\mu_2}{2}\left\|\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right)\right\|_2^2 \\ & + \frac{\mu_3}{2}\left\|s\mathbf{1}_u - Y\left(X_u\beta + \beta_0\mathbf{1}_u\right)\right\|_2^2 \\ & + \frac{\mu_4}{2}\left\|\beta - \mathbf{t}\right\|_2^2 \\ & + \frac{\mu_5}{2}\left\|r - \frac{1}{u}\sum_{i=l+1}^{n}\left(x_i\beta + \beta_0\right)\right\|_2^2, \end{aligned} \tag{11}$$

where $\mu_1, \mu_2, \mu_3, \mu_4 > 0$ are parameters. Problem (11) is the form of ADMM, which consists of the following iterations:

$$\begin{aligned} & \left(\beta^{k+1}, \beta_0^{k+1}, \mathbf{h}^{k+1}, \mathbf{a}^{k+1}, \mathbf{b}^{k+1}, \mathbf{t}^{k+1}\right) \\ & \quad = \arg\min_{\beta,\beta_0,\mathbf{h},\mathbf{a},\mathbf{b},\mathbf{t}} L\left(\beta, \beta_0, \mathbf{h}, \mathbf{a}, \mathbf{b}, \mathbf{t}, \mathbf{u}_h^k, \mathbf{u}_a^k, \mathbf{u}_b^k, \mathbf{v}^k, q^k\right), \\ & \mathbf{u}_h^{k+1} = \mathbf{u}_h^k + \mu_1\left(\mathbf{1}_l - Y_l\left(X_l\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_l\right) - \mathbf{h}^{k+1}\right), \\ & \mathbf{u}_a^{k+1} = \mathbf{u}_a^k + \mu_2\left(\mathbf{1}_u - Y_u\left(X_u\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_u\right) - \mathbf{a}^{k+1}\right), \\ & \mathbf{u}_b^{k+1} = \mathbf{u}_b^k + \mu_3\left(s\mathbf{1}_u - Y_u\left(X_u\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_u\right) - \mathbf{b}^{k+1}\right), \\ & \mathbf{v}^{k+1} = \mathbf{v}^k + \mu_4\left(\beta^{k+1} - t^{k+1}\right), \\ & q^{k+1} = q^k + \mu_5\left(r - \frac{1}{u}\sum_{i=l+1}^{n}\left(x_i\beta^{k+1} + \beta_0^{k+1}\right)\right). \end{aligned} \tag{12}$$

The efficiency of the iterative algorithm (12) lies on whether the first equation of (12) can be solved quickly. According to the theory of ADMM, these variables (β, β_0), $\mathbf{h}$, $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{t}$ are updated in an alternating or sequential fashion, which accounts for the term alternating direction. So we can get

$$\begin{aligned} & \left(\beta^{k+1}, \beta_0^{k+1}\right) = \arg\min_{\beta,\beta_0} L\left(\beta, \beta_0, \mathbf{h}^k, \mathbf{a}^k, \mathbf{b}^k, \mathbf{t}^k, \mathbf{u}_h^k, \mathbf{u}_a^k, \mathbf{u}_b^k, \mathbf{v}^k, q^k\right), \\ & \mathbf{h}^{k+1} = \arg\min_{\mathbf{h}} L\left(\beta^{k+1}, \beta_0^{k+1}, \mathbf{h}, \mathbf{a}^k, \mathbf{b}^k, \mathbf{t}^k, \mathbf{u}_h^k, \mathbf{u}_a^k, \mathbf{u}_b^k, \mathbf{v}^k, q^k\right), \\ & \mathbf{a}^{k+1} = \arg\min_{\mathbf{a}} L\left(\beta^{k+1}, \beta_0^{k+1}, \mathbf{h}^{k+1}, \mathbf{a}, \mathbf{b}^k, \mathbf{t}^k, \mathbf{u}_h^k, \mathbf{u}_a^k, \mathbf{u}_b^k, \mathbf{v}^k, q^k\right), \\ & \mathbf{b}^{k+1} = \arg\min_{\mathbf{b}} L\left(\beta^{k+1}, \beta_0^{k+1}, \mathbf{h}^{k+1}, \mathbf{a}^{k+1}, \mathbf{b}, \mathbf{t}^k, \mathbf{u}_h^k, \mathbf{u}_a^k, \mathbf{u}_b^k, \mathbf{v}^k, q^k\right), \\ & \mathbf{t}^{k+1} = \arg\min_{\mathbf{t}} L\left(\beta^{k+1}, \beta_0^{k+1}, \mathbf{h}^{k+1}, \mathbf{a}^{k+1}, \mathbf{b}^{k+1}, \mathbf{t}, \mathbf{u}_h^k, \mathbf{u}_a^k, \mathbf{u}_b^k, \mathbf{v}^k, q^k\right). \end{aligned} \tag{13}$$

For the first equation in (13), it is equivalent to the following convex optimization:

$$\begin{aligned} & \left(\beta^{k+1}, \beta_0^{k+1}\right) \\ & \quad = \arg\min_{\beta,\beta_0} \frac{\lambda_2}{2}\|\beta\|_2^2 + \mathbf{u}_h^T\left(\mathbf{1}_l - Y_l\left(X_l\beta + \beta_0\mathbf{1}_l\right) - \mathbf{h}\right) \\ & \qquad + \mathbf{u}_a^T\left(\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) - \mathbf{a}\right) \\ & \qquad + \mathbf{u}_b^T\left(s\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) - \mathbf{b}\right) \\ & \qquad + \mathbf{v}^T(\beta - \mathbf{t}) + q\left(r - \frac{1}{u}\sum_{i=1}^{u}\left(x_i\beta + \beta_0\right)\right) \\ & \qquad + \frac{\mu_1}{2}\left\|\mathbf{1}_l - Y_l\left(X_l\beta + \beta_0\mathbf{1}_l\right) - \mathbf{h}\right\|_2^2 \\ & \qquad + \frac{\mu_2}{2}\left\|\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) - \mathbf{a}\right\|_2^2 \\ & \qquad + \frac{\mu_3}{2}\left\|s\mathbf{1}_u - Y_u\left(X_u\beta + \beta_0\mathbf{1}_u\right) - \mathbf{b}\right\|_2^2 \\ & \qquad + \frac{\mu_4}{2}\left\|\beta - \mathbf{t}\right\|_2^2 \\ & \qquad + \frac{\mu_5}{2}\left\|r - \frac{1}{u}\sum_{i=l+1}^{n}\left(x_i\beta + \beta_0\right)\right\|_2^2. \end{aligned} \tag{14}$$

The objective function in the above minimization problem is quadratic and differentiable, and since $(\beta^{k+1}, \beta_0^{k+1})$ minimizes this function by definition, the optimal solution can be found by solving a set of linear equations

$$\begin{pmatrix} \left(\lambda_2+\mu_4+\dfrac{\mu_5}{u^2}\sum_{i=l+1}^{n}x_i^T\sum_{i=l+1}^{n}x_i\right)\mathbf{I}+\mu_1X_l^TX_l+(\mu_2+\mu_3)X_u^TX_u & \mu_1X_l^T\mathbf{1}_l+(\mu_2+\mu_3)X_u^T\mathbf{1}_u+\dfrac{\mu_5}{u}\sum_{i=l+1}^{n}x_i^T \\ \mathbf{1}_l^TX_l+(\mu_2+\mu_3)\mathbf{1}_u^TX_u+\dfrac{\mu_5}{u}\sum_{i=l+1}^{n}x_i\mathbf{1}_u & l\mu_1+2u(\mu_2+\mu_3)+\mu_5 \end{pmatrix}\begin{pmatrix}\beta^{k+1}\\ \beta_0^{k+1}\end{pmatrix} \tag{15}$$
$$= \begin{pmatrix} X_l^TY_l^T\left(\mathbf{u}_h^k+\mu_1\left(1-\mathbf{h}^k\right)\right)+X_u^TY_u^T\left(\mathbf{u}_a^k+\mathbf{u}_b^k+\mu_2\left(1-\mathbf{a}^k\right)+\mu_3\left(s-\mathbf{b}^k\right)\right)-\mathbf{v}^k+\left(q^k+\mu_5r\right)\sum_{i=l+1}^{n}\dfrac{x_i^T}{u}+\mu_4\mathbf{t}^k \\ \mathbf{1}_l^TY_l^T\left(\mathbf{u}_h^k+\mu_1\left(1-\mathbf{h}^k\right)\right)+\mathbf{1}_u^TY_u^T\left(\mathbf{u}_a^k+\mathbf{u}_b^k+\mu_2\left(1-\mathbf{a}^k\right)+\mu_3\left(s-\mathbf{b}^k\right)\right)+q^k+\mu_5r \end{pmatrix}.$$

In (15), $\mathbf{I}$ is a $p\times p$ unit matrix and the coefficient matrix is a $(p+1)\times(p+1)$ matrix, independent of the optimization variables. For large p, small n setting, the term $X_l^TX_l$ in the coefficient matrix will be a positive low rank matrix with rank at most l while the term $X_u^TX_u$ in the coefficient matrix will be a positive low rank matrix with rank at most $2u$. Therefore, the coefficient matrix is also low rank matrix with rank at most $(2u+1)$. And if we use CG to solve the problem (15), it will converge in less than $(2u+1)$ steps [21].

For the second equation in (13), it is equivalent to solving

$$\begin{aligned}\mathbf{h}^{k+1} = {} & \arg\min_{\mathbf{h}} C\sum_{i=1}^{l}(h_i)_+ \\ & +\left(\mathbf{u}_h^k\right)^T\left(\mathbf{1}_l-Y_l\left(X_l\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_l\right)-\mathbf{h}^{k+1}\right) \\ & +\frac{\mu_1}{2}\left\|\mathbf{1}_l-Y_l\left(X_l\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_l\right)-\mathbf{h}^{k+1}\right\|_2^2.\end{aligned} \tag{16}$$

In order to solve (16), we need the following Proposition [22].

Proposition 2. *Let* $s_\lambda(w)=\arg\min_{x\in\mathbb{R}}\lambda x_+ + 0.5\|x-w\|_2^2$ *where* $\lambda>0$. *Then*

$$s_\lambda(w)=\begin{cases} w-\lambda & w>\lambda, \\ 0 & 0\le w\le\lambda, \\ w & w<0. \end{cases} \tag{17}$$

Combined with Proposition 2 and

$$\begin{aligned}&\frac{\left\|\mathbf{u}_h^k\right\|_2^2}{2\mu_1}+\left(\mathbf{u}_h^k\right)^T\left(\mathbf{1}_l-Y_l\left(X_l\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_l\right)-\mathbf{h}^{k+1}\right) \\ &\quad+0.5\mu_1\left\|\mathbf{1}_l-Y_l\left(X_l\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_l\right)-\mathbf{h}^{k+1}\right\|_2^2 \\ &=0.5\mu_1\left\|\frac{\mathbf{u}_h^k}{\mu_1}+\mathbf{1}_l-Y_l\left(X_l\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_l\right)-\mathbf{h}^{k+1}\right\|_2^2,\end{aligned} \tag{18}$$

we can update $\mathbf{h}^{k+1}$ according to Corollary 3.

Corollary 3. *The update of* $\mathbf{h}^{k+1}$ *in* (16) *is equivalent to*

$$\mathbf{h}^{k+1}=S_{C/\mu_1}\left(\mathbf{1}_l+\frac{\mathbf{u}_h^k}{\mu_1}-Y_l\left(X_l\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_l\right)\right), \tag{19}$$

where $S_\lambda(w)=(s_\lambda(w_1),\ldots,s_\lambda(w_l))$.

For the third equation in (13), it is equivalent to solving (20):

$$\begin{aligned}\mathbf{a}^{k+1} = {} & \arg\min_{\mathbf{a}} C^*\sum_{i=l+1}^{l+2u}(a_i)_+ \\ & +\left(\mathbf{u}_a^k\right)^T\left(\mathbf{1}_u-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)-\mathbf{a}^{k+1}\right) \\ & +\frac{\mu_2}{2}\left\|\mathbf{1}_u-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)-\mathbf{a}^{k+1}\right\|_2^2.\end{aligned} \tag{20}$$

Combined with Proposition 2 and

$$\begin{aligned}&\frac{\left\|\mathbf{u}_a^k\right\|_2^2}{2\mu_1}+\left(\mathbf{u}_a^k\right)^T\left(\mathbf{1}_u-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)-\mathbf{a}^{k+1}\right) \\ &\quad+0.5\mu_2\left\|\mathbf{1}_u-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)-\mathbf{a}^{k+1}\right\|_2^2 \\ &=0.5\mu_2\left\|\frac{\mathbf{u}_a^k}{\mu_2}+\mathbf{1}_u-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)-\mathbf{a}^{k+1}\right\|_2^2,\end{aligned} \tag{21}$$

we can update $\mathbf{a}^{k+1}$ according to Corollary 4.

Corollary 4. *The update of* $\mathbf{a}^{k+1}$ *in* (20) *is equivalent to*

$$\mathbf{a}^{k+1}=S_{C^*/\mu_2}\left(\mathbf{1}_u+\frac{\mathbf{u}_a^k}{\mu_2}-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)\right), \tag{22}$$

where $S_\lambda(w)=(s_\lambda(w_1),\ldots,s_\lambda(w_{2u}))$.

For the fourth equation in (13), it is equivalent to solving

$$\begin{aligned}\mathbf{b}^{k+1} = {} & \arg\min_{\mathbf{b}} C^*\sum_{i=l+1}^{l+2u}(b_i)_+ \\ & +\left(\mathbf{u}_b^k\right)^T\left(s\mathbf{1}_u-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)-\mathbf{b}^{k+1}\right) \\ & +\frac{\mu_3}{2}\left\|s\mathbf{1}_u-Y_u\left(X_u\beta^{k+1}+\beta_0^{k+1}\mathbf{1}_u\right)-\mathbf{b}^{k+1}\right\|_2^2.\end{aligned} \tag{23}$$

In order to solve (23), we need the following proposition.

Proposition 5. *Let* $g_\lambda(w)=\arg\min_{x\in\mathbb{R}}\lambda x_+ + 0.5\|x-w\|_2^2$, *where* $\lambda<0$. *Then*

$$g_\lambda(w)=\begin{cases} w-\lambda & w>0, \\ w & w<0. \end{cases} \tag{24}$$

Proof. The function $\lambda x_+ + 0.5\|x-w\|_2^2$ is strongly convex, so it has a unique solution. Therefore, by the subdifferential calculus [23], g_λ is the unique minimizer of the following equation:

$$0 \in \lambda \partial (x_+) + x - w, \tag{25}$$

where $\partial(x_+)$ is the subdifferential of the function x_+. According to [23], $\partial(x_+)$ can be expressed as

$$\partial (x_+) = \begin{cases} 1 & x > 0, \\ \{p : p \in [0,1]\} & x = 0, \\ 0 & x < 0. \end{cases} \tag{26}$$

With (25) and (26), we can get the desired result.
Then, combined with Proposition 5 and

$$\begin{aligned} &\frac{\left\|\mathbf{u}_b^k\right\|_2^2}{2\mu_3} + \left(\mathbf{u}_b^k\right)^T \left(s\mathbf{1}_u - Y_u\left(X_u\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_u\right) - \mathbf{b}^{k+1}\right) \\ &\quad + 0.5\mu_3 \left\|s\mathbf{1}_u - Y_u\left(X_u\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_u\right) - \mathbf{b}^{k+1}\right\|_2^2 \\ &= 0.5\mu_3 \left\|\frac{\mathbf{u}_b^k}{\mu_3} + s\mathbf{1}_u - Y_u\left(X_u\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_u\right) - \mathbf{b}^{k+1}\right\|_2^2, \end{aligned} \tag{27}$$

we can update $\mathbf{b}^{k+1}$ according to Corollary 6. □

Corollary 6. *The update of* $\mathbf{b}^{k+1}$ *in* (23) *is equivalent to*

$$\mathbf{b}^{k+1} = G_{-C^*/\mu_3}\left(s\mathbf{1}_u + \frac{\mathbf{u}_b^k}{\mu_3} - Y_u\left(X_u\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_u\right)\right), \tag{28}$$

where $G_\lambda(w) = (g_\lambda(w_1), \ldots, g_\lambda(w_{2u}))$.

For the fifth equation in (13), it is equivalent to solving

$$\begin{aligned} \mathbf{t}^{k+1} = \ &\arg\min_{\mathbf{t}} \lambda_1 \left\|\mathbf{t}^{k+1}\right\|_1 + \left(\mathbf{v}^k\right)^T \left(\beta^{k+1} - \mathbf{t}^{k+1}\right) \\ &+ 0.5\mu_4 \left\|\beta^{k+1} - \mathbf{t}^{k+1}\right\|_2^2. \end{aligned} \tag{29}$$

Solving (29) can be done efficiently using soft threshold, and we can update $\mathbf{t}^{k+1}$ according to Corollary 7.

Corollary 7. *The update of* $\mathbf{t}^{k+1}$ *in* (29) *is*

$$\mathbf{t}^{k+1} = T_{\lambda_1/\mu_4}\left(\frac{\mathbf{v}^k}{\mu_4} + \beta^{k+1}\right), \tag{30}$$

where $T_\lambda(w) = (t_\lambda(w_1), \ldots, t_\lambda(w_p))$, *and* $t_\lambda(w) = \operatorname{sgn}(w)\max\{0, |w| - \lambda\}$.

According to the theory of ADMM, $\|s_1^k\|_2$, $\|s_2^k\|_2$, $\|s_3^k\|_2$, and $\|s_4^k\|_2$ must be small below some certain threshold δ. Considering the auxiliary variables in (11), we expect that $\|\mathbf{1}_l - Y_l(X_l\beta + \beta_0\mathbf{1}_l) - \mathbf{h}\|_2$, $\|\mathbf{1}_u - Y_u(X_u\beta + \beta_0\mathbf{1}_u) - \mathbf{a}\|_2$, $\|s\mathbf{1}_u - Y_u(X_u\beta + \beta_0\mathbf{1}_u) - \mathbf{b}\|_2$, and $\|\beta - \mathbf{t}\|_2$ also must be small. Therefore, in our experiment, algorithm for SENFS stops whenever

$$\begin{gathered} \left\|s_1^k\right\|_2 < \delta, \quad \left\|s_2^k\right\|_2 < \delta, \quad \left\|s_3^k\right\|_2 < \delta, \quad \left\|s_4^k\right\|_2 < \delta, \\ \left\|\mathbf{1}_l - Y_l(X_l\beta + \beta_0\mathbf{1}_l) - \mathbf{h}\right\|_2 < \sqrt{l}\delta, \\ \left\|\mathbf{1}_u - Y_u(X_u\beta + \beta_0\mathbf{1}_u) - \mathbf{a}\right\|_2 < \sqrt{2u}\delta, \\ \left\|s\mathbf{1}_u - Y_u(X_u\beta + \beta_0\mathbf{1}_u) - \mathbf{b}\right\|_2 < \sqrt{2u}\delta, \\ \left\|\beta - \mathbf{t}\right\|_2 < \sqrt{p}\delta. \end{gathered} \tag{31}$$

Finally, we can get the algorithm for SENFS. The detailed procedure of the algorithm ADMM for SENFS is summarized in Algorithm 8 as follows.

Algorithm 8. ADMM algorithm for SENFS.

Input. Labeled data set $\{x_i, y_i\}_{i=1}^l$; unlabeled data set $\{x_i\}_{i=l+1}^n$; tuning parameters λ_1 and λ_2; regularization parameter C.

Output. Selected feature set.

Step 1. Initialize β^0, β_0^0, $\mathbf{h}^0$, $\mathbf{a}^0$, $\mathbf{b}^0$, $\mathbf{t}^0$, $\mathbf{u}_h^0$, $\mathbf{u}_a^0$, $\mathbf{u}_b^0$, $\mathbf{v}^0$, and q^0.

Step 2. If (31) is satisfied, go to Step 3; otherwise,

(1) update β^{k+1} and β_0^{k+1} according to (15);

(2) update $\mathbf{h}^{k+1}$, $\mathbf{a}^{k+1}$, $\mathbf{b}^{k+1}$, and $\mathbf{t}^{k+1}$ according to Corollaries 3–7, respectively;

(3) update $\mathbf{u}_h^{k+1}$, $\mathbf{u}_a^{k+1}$, $\mathbf{u}_b^{k+1}$, $\mathbf{v}^{k+1}$, and q^{k+1} according to (12).

Step 3. Get the best feature subset according to β_j^{k+1}, $j = 1, \ldots, p$. If $\beta_j^{k+1} = 0$, the corresponding jth feature is abandoned; otherwise, it is selected as an important feature.

3.2. Convergence Analysis and Computational Cost. The convergence property of Algorithm 8 can be derived from the theory of the alternating direction method of multipliers. According to the standard convergence theory of ADMM, Algorithm 8 satisfies the dual variable convergence [24]. So Theorem 9 holds.

Theorem 9. *Suppose that* $(\beta^*, \beta_0{}^*)$ *is one of solution of* (5). *Then the following property holds:*

$$\begin{aligned} &C\sum_{i=1}^{l} H_1\left(y_i\left(x_i\beta^k + \beta_0^k\right)\right) \\ &\quad + C^*\sum_{i=l+1}^{l+2u}\left(H_1\left(y_i\left(x_i\beta^k + \beta_0^k\right)\right) - H_s\left(y_i\left(x_i\beta^k + \beta_0^k\right)\right)\right) \\ &\quad + \lambda_1\left\|\beta^k\right\|_1 + \frac{\lambda_2\left\|\beta^k\right\|_2^2}{2} \end{aligned}$$

$$= C\sum_{i=1}^{l} H_1\left(y_i\left(x_i\beta^* + \beta_0^*\right)\right) + C^*\sum_{i=l+1}^{l+2u}\left(H_1\left(y_i\left(x_i\beta^* + \beta_0^*\right)\right) - H_s\left(y_i\left(x_i\beta^* + \beta_0^*\right)\right)\right) + \lambda_1\left\|\beta^*\right\|_1 + \frac{\lambda_2\left\|\beta^*\right\|_2^2}{2}. \tag{32}$$

As for the computational issue, it is hard to predict the computational cost because it depends on the all the penalty parameters. According to our experience, we only need to iterate a few hundred iterations to get a reasonable result. On the other hand, the efficiency of Algorithm 8 lies mainly on whether we can quickly solve the linear equations (15). And the computational cost for solving (15) is $O(l^2p + 4u^2p)$.

3.3. Varying Penalty Parameter. In order to make performance less dependent on the initial choice of the penalty parameter, it is necessary to use different penalty parameters. According to our experiment experience, the penalty parameters μ_1, μ_2, μ_3, and μ_4 have a huge influence on the performance and the number of iterations involved, so adaptive selections of them are performed.

For μ_1, let $f(\beta,\beta_0) = C\sum_{i=1}^{l}(h_i)_+ + C^*\sum_{i=l+1}^{l+2u}(H_1(y_i(x_i\beta+\beta_0)) - H_s(y_i(x_i\beta+\beta_0))) + \lambda_1\|\beta\|_1 + 0.5\lambda_2\|\beta\|_2^2$, and the constraint conditions are $\mathbf{h} = \mathbf{1}_l - Y_l(X_l\beta + \beta_0\mathbf{1}_l)$ and the constraint of (3). The optimization task (6) is equivalent to

$$\begin{aligned} W(\beta,\beta_0) = {} & f(\beta,\beta_0) + \mathbf{u}_h^T\left(\mathbf{1}_l - Y_l\left(X_l\beta + \beta_0\mathbf{1}_l\right) - \mathbf{h}\right) \\ & + q\left(r - \frac{1}{u}\sum_{i=l+1}^{n}\left(x_i\beta + \beta_0\right)\right) \\ & + \frac{\mu_1}{2}\left\|\mathbf{1}_l - Y_l\left(X_l\beta + \beta_0\mathbf{1}_l\right) - \mathbf{h}\right\|_2^2 \\ & + \frac{\mu_5}{2}\left\|r - \frac{1}{u}\sum_{i=l+1}^{n}\left(x_i\beta + \beta_0\right)\right\|_2^2. \end{aligned} \tag{33}$$

The necessary optimality conditions for the problem (6) are dual feasibility as

$$0 \in \frac{\partial f\left(\beta^*,\beta_0^*\right)}{\partial\beta^*} - X_l^TY_l^T\mathbf{u}_h^* - \frac{q^*}{u}\sum_{i=l+1}^{n}x_i^T. \tag{34}$$

Since β^{k+1} minimizes $W(\beta^{k+1},\beta_0^{k+1},\mathbf{u}_h^k,q^k)$ by definition, we have that

$$\begin{aligned} 0 \in {} & \frac{\partial f\left(\beta^{k+1},\beta_0^{k+1}\right)}{\partial\beta^{k+1}} \\ & - X_l^TY_l^T\left(\mathbf{u}_h^k - \mu_1\left(\mathbf{1}_l - Y_l\left(X_l\beta^{k+1} + \beta_0^{k+1}\mathbf{1}_l\right) - \mathbf{h}^k\right)\right) \\ & - \frac{q^k}{u}\sum_{i=l+1}^{n}x_i^T - \frac{\mu_5}{u} \\ & \times \sum_{i=l+1}^{n}x_i^T\left(r - \frac{1}{u}\sum_{i=l+1}^{n}\left(x_i\beta^{k+1} + \beta_0^{k+1}\right)\right) \\ = {} & \frac{\partial f\left(\beta^{k+1},\beta_0^{k+1}\right)}{\partial\beta^{k+1}} - X_l^TY_l^T\mathbf{u}_h^{k+1} - \frac{q^{k+1}}{u} \\ & \times \sum_{i=l+1}^{n}x_i^T + \mu_1X_l^TY_l^T\left(\mathbf{h}^k - \mathbf{h}^{k+1}\right). \end{aligned} \tag{35}$$

Compared with (34), (35) means that the quantity $\mu_1X_l^TY_l^T(\mathbf{h}^k - \mathbf{h}^{k+1})$ can be viewed as a residual for (34), and let $s_1^{k+1} = \mu_1X_l^TY_l^T(\mathbf{h}^k - \mathbf{h}^{k+1})$.

For μ_2, let $f(\beta,\beta_0) = C\sum_{i=1}^{l}H_1(y_i(x_i\beta+\beta_0)) + C^*\sum_{i=l+1}^{l+2u}((a_i)_+ - H_s(y_i(x_i\beta+\beta_0))) + \lambda_1\|\beta\|_1 + 0.5\lambda_2\|\beta\|_2^2$, and the constraint conditions are $\mathbf{a} = \mathbf{1}_u - Y_u(X_u\beta + \beta_0\mathbf{1}_u)$ and the constraint of (3). Through the same solving process as parameter μ_1, we can get the residual $s_2^{k+1} = \mu_2^{k+1}X_u^TY_u^T(\mathbf{a}^k - \mathbf{a}^{k+1})$. Similarly, we can get the residual $s_3^{k+1} = \mu_3^{k+1}X_u^TY_u^T(\mathbf{b}^k - \mathbf{b}^{k+1})$ for parameter μ_3 and $s_4^{k+1} = \mu_4^{k+1}X_u^TY_u^T(\mathbf{t}^{k+1} - \mathbf{t}^k)$ for parameter μ_4. With these residuals, we can get a simple scheme to update μ_1,μ_2,μ_3, and μ_4, respectively, according to Corollary 10.

Corollary 10. *The update of μ_1, μ_2, μ_3, μ_4 is*

$$\mu_1^{k+1} = \begin{cases} \tau^{incr}\mu_1^k & \text{if } \left\|\mathbf{1}_l - Y_l(X_l\beta^k + \beta_0^k\mathbf{1}_l) - \mathbf{h}^k\right\|_2 > \theta\left\|s_1^k\right\|_2 \\ \dfrac{\mu_1^k}{\tau^{decr}} & \text{if } \left\|s_1^k\right\|_2 > \theta\left\|\mathbf{1}_l - Y_l(X_l\beta^k + \beta_0^k\mathbf{1}_l) - \mathbf{h}^k\right\|_2 \\ \mu_1^k & \text{otherwise,} \end{cases}$$

$$\mu_2^{k+1} = \begin{cases} \tau^{incr}\mu_2^k & \text{if } \left\|\mathbf{1}_u - Y_u(X_u\beta^k + \beta_0^k\mathbf{1}_u) - \mathbf{a}^k\right\|_2 > \theta\left\|s_2^k\right\|_2 \\ \dfrac{\mu_2^k}{\tau^{decr}} & \text{if } \left\|s_2^k\right\|_2 > \theta\left\|\mathbf{1}_u - Y_u(X_u\beta^k + \beta_0^k\mathbf{1}_u) - \mathbf{a}^k\right\|_2 \\ \mu_2^k & \text{otherwise,} \end{cases}$$

$$\mu_3^{k+1} = \begin{cases} \tau^{incr}\mu_3^k & \text{if } \left\|s\mathbf{1}_u - Y_u(X_u\beta^k + \beta_0^k\mathbf{1}_u) - \mathbf{b}^k\right\|_2 > \theta\left\|s_3^k\right\|_2 \\ \dfrac{\mu_3^k}{\tau^{decr}} & \text{if } \left\|s_3^k\right\|_2 > \theta\left\|s\mathbf{1}_u - Y_u(X_u\beta^k + \beta_0^k\mathbf{1}_u) - \mathbf{b}^k\right\|_2 \\ \mu_3^k & \text{otherwise,} \end{cases}$$

$$\mu_4^{k+1} = \begin{cases} \tau^{incr}\mu_4^k & \text{if } \left\|\beta - \mathbf{t}\right\|_2 > \theta\left\|s_4^k\right\|_2 \\ \dfrac{\mu_4^k}{\tau^{decr}} & \text{if } \left\|s_4^k\right\|_2 > \theta\left\|\beta - \mathbf{t}\right\|_2 \\ \mu_4^k & \text{otherwise,} \end{cases} \tag{36}$$

where $\tau^{incr} > 1$, $\tau^{decr} >$, and $\theta > 1$ are parameters. Typical choices might be $\tau^{incr} = \tau^{decr} = 2$ and $\theta = 10$.

4. Experimental Evaluation

This section examines the performance of SENFS with respect to its feature selection and test error on simulated

Table 1: Comparisons of selected features and their standard errors (in parenthesis). F_{relevant} is the number of selected relevant features and F_{noise} is the number of selected noise features.

ρ	SENFS		Spectral		DrSVM	
	F_{relevant}	F_{noise}	F_{relevant}	F_{noise}	F_{relevant}	F_{noise}
0	6.2 (0.28)	5.6 (0.28)	4.75 (0.22)	5.9 (0.29)	5.41 (0.27)	5.7 (0.15)
0.5	7.2 (0.01)	3.9 (0.05)	6.88 (0.1)	6.2 (0.24)	6.83 (0.31)	5.4 (0.17)
0.9	8.9 (0.04)	2.1 (0.11)	8.32 (0.09)	4.15 (0.15)	8.1 (0.08)	4.55 (0.22)

Table 2: Comparisons of test errors, computational time needed, and their standard errors (in parenthesis).

d	SENFS		Spectral		DrSVM	
	Test error	Time	Test error	Time	Test error	Time
1	9.4 (0.12)%	6.7 (1.1)	15.7 (0.27)%	5.5 (1.0)	17.1 (0.31)%	6.1 (1.2)
2	8.7 (0.13)%	6.4 (0.2)	14.2 (0.14)%	5.3 (0.1)	16.2 (0.27)%	5.8 (0.2)
3	7.5 (0.02)%	5.9 (0.4)	12.3 (0.07)%	4.7 (0.1)	14.6 (0.11)%	4.9 (0.2)

Table 3: Data sets used in the experiments. n, p, and l are the number of samples, features, and labeled samples, respectively.

Data set	n	p	l
Digit1	1500	241	10 or 100
USPS	1500	241	10 or 100
COIL2	1500	241	10 or 100
BCI	400	117	10 or 100
g241c	1500	241	10 or 100
g241n	1500	241	10 or 100

data and six benchmark data sets. In order to evaluate the effectiveness of SENFS, we compare SENFS with an existing semisupervised feature selection algorithm: Spectral [10], and a supervised feature selection algorithm: DrSVM [11], which also has the characteristics of grouping effect. On the other hand, in order to evaluate the quality of selected features, SVM was executed on these selected features. The experiments are run on a desktop with Pentium(R) 2.0 G CPU, 1.99 G main memory. The programs are compiled in Windows system with Matlab in version R2009a.

The limited number of samples prohibits having enough and independent training and testing data for performance evaluation. It is very common to apply accross-validation (CV) in this scenario. We used 5-fold CV: we partitioned the data set into five complementary subsets of equal size. Four subsets were used as training data; the remaining subset served as test data. We repeated this process five times such that each of the five subsets was used exactly once as test data. To get more reliable estimate, we performed the 5-fold CV for 10 times and the experimental results are average results over test data sets. Moreover, finding the appropriate value of the tuning parameter pair λ_1 and λ_2 is essential for the performance of SENFS. We employed 10-fold CV over a large grid.

4.1. Simulation. We evaluate the performance of SENFS using two parameters: the correlation between relevant features denoted by ρ, the number of labeled samples, and the degree of overlapping among classes denoted by d. Consider 2-class problem in which the samples are lying in a p dimensional space with the first 10 dimensional being relevant to classification and the remaining features being noise, where the correlation between the first 10 features is ρ. The number of samples is 300 with $p = 500$. For the samples from +1 class, they are sampled from a normal distribution with mean and covariance as follows:

$$\sum = \begin{pmatrix} \overset{*}{\underset{10\times 10}{\sum}} & 0_{10\times(p-10)} \\ 0_{(p-10)\times 10} & \mathbf{I}_{(p-10)\times(p-10)} \end{pmatrix}, \tag{37}$$

where the diagonal elements of $\sum^*$ are 1 and the off-diagonal elements are all equal to ρ. The -1 class has a similar distribution expect that its mean is:

$$\alpha_- = \left(\underbrace{-d, \ldots, -d}_{10}, \underbrace{0, \ldots, 0}_{p-10} \right). \tag{38}$$

To evaluate the effect of the correlation between relevant features, SENFS is compared with Spectral and DrSVM, measured by the number of selected features with two labeled samples and $d = 1$. The results are summarized in Table 1. As shown in Table 1, on this simulated data, when the relevant features are highly correlated (e.g., $\rho = 0.9$), Spectral and DrSVM tend to keep only a small subset of the relevant correlated variables and overlook the others, while the SENFS tends to identify all of them, due to the grouping effect. These three methods seem to work well in removing irrelevant features.

The effects of the number of labeled samples on test error over the top 10 selected features are summarized in Figure 1 with $d = 1$ and $\rho = 0.9$. As can be seen, the test errors of SENFS, Spectral, and DrSVM decrease with the increase of the number of labeled samples, but SENFS seems to achieve the best classification performance when the number of labeled samples is varying, which may imply that SENFS can make better use of the labeled samples than spectral and

Table 4: Comparisons of test errors and their standard errors (in parenthesis) on benchmark data sets.

Dataset	SENFS		Spectral		DrSVM	
	$l = 10$	$l = 100$	$l = 10$	$l = 100$	$l = 10$	$l = 100$
Digit1	15.1 (0.04)%	4.28 (0.12)%	17.7 (0.09)%	8.7 (0.15)%	21.35 (0.05)%	5.85 (0.16)%
USPS	21.17 (0.03)%	11.1 (0.24)%	29.87 (0.15)%	12.6 (0.31)%	31.8 (0.12)%	17.3 (0.3)%
COIL2	36.1 (1.04)%	13.6 (0.98)%	37.7 (1.3)%	16.9 (0.91)%	47.2 (0.8)%	31.2 (1.1)%
BCI	43.9 (0.23)%	33.8 (0.24)%	49.87 (0.15)%	42.1 (0.31)%	47.8 (0.12)%	36.7 (0.8)%
g241c	32.8 (1.14)%	24.9 (1.12)%	42.8 (1.29)%	32.7 (1.5)%	41.75 (1.1)%	26.0 (1.6)%
g241n	37.6 (0.83)%	24.3 (0.24)%	45.7 (0.3)%	33.6 (0.51)%	41.8 (0.32)%	25.7 (0.67)%

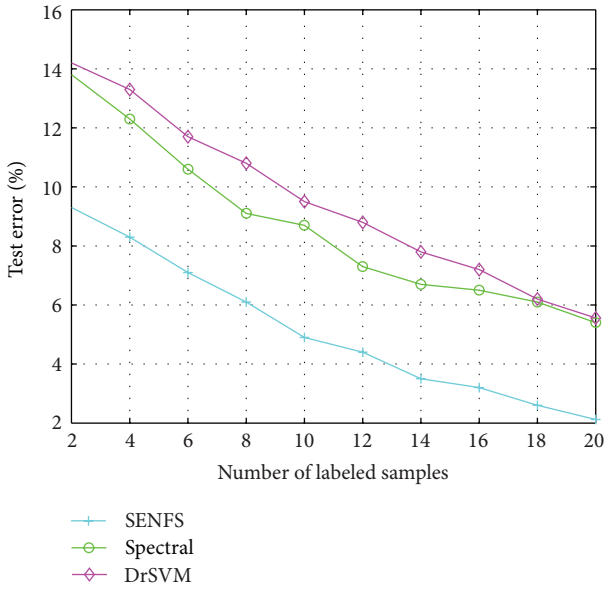

Figure 1: Test error versus the number of labeled samples.

DrSVM. The supervised feature selection method DrSVM achieves the worst results because it only relies on the few labeled samples and discards the large amount of unlabeled samples.

In Table 2, the effect of the degree of overlapping among classes on test error over the top 5 selected features is evaluated with two labeled samples and $d = 1$ and $\rho = 0.9$, also reporting the typical computational time of our experimental campaign. As we can see, SENFS seems to have the best prediction performance. When d is small, the two classes overlap largely and in this case, other methods achieved worse performance compared with SENFS. However, SENFS, solving the programming problem (6) needing an iterative procedure, requires more computational time than the other methods as you can see in Table 2. It is noted that the absolute values are not as important as the relative differences between the individual methods.

4.2. Application to Benchmark Data Sets. Several benchmark data sets are selected to test the performance of SENFS, which are used as benchmark data sets in [7, 8] to test the performances of semisupervised algorithms. These benchmark data sets consist of 9 semisupervised learning data sets. We did not test the SSL6, SSL8, and SSL9 data sets since the SSL6 data set includes six classes, the SSL8 data set contains too many samples (n is over one million) and the SSL9 data set has too many dimensions (p is over ten thousand). The names and characteristics of the left six data sets are given in Table 3.

In this study, we examine performance evaluation through 5-fold cross-validation that is, we randomly select four fifths of the unlabeled samples, plus all the labeled samples, for SENFS, Spectral, and DrSVM to select optimal feature subsets, while leaving the remaining one fifth for testing test error on the selected features using SVM, where all the labeled samples are used for training SVM. The results measured by test error are reported in Table 4. As can be seen, SENFS outperforms the semisupervised and supervised feature selection methods on all the six data sets when $l = 10$ and $l = 100$. When $l = 10$, Spectral performs the second best, on USPS, COIL2, and BCI data sets, while DrSVM performs the second best on Digit1, BCI, g241c and g241n data sets when $l = 100$.

5. Conclusion

This paper has proposed a novel semisupervised feature selection algorithm based on SVM and the elastic net penalty. The whole methodology of SENFS and the solution path based on ADMM have been described in detail in this paper. The experimental results illustrate that SENFS can identify the relevant features and encourage highly correlated features to be selected or removed together.

Future work will address how these selected features interpret their semantic relationship with the data they are selected from, which can be used for unknown data analysis, and extend SENFS to be suitable for multiclass case.

Appendix

Proof of Theorem 1. Consider another set of coefficients

$$\widetilde{\beta}_0^* = \widetilde{\beta}_0, \qquad \widetilde{\beta}_{j'}^* = \begin{cases} \dfrac{\widetilde{\beta}_j + \widetilde{\beta}_k}{2}, & \text{if } j' = j \text{ or } j' = k, \\ \widetilde{\beta}_{j'} & \text{otherwise.} \end{cases} \tag{A.1}$$

Then we have

$$
\begin{aligned}
&C\sum_{i=1}^{l} H_1\left(y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right)\right) \\
&\quad + C^*\sum_{i=l+1}^{l+2u}\left(H_1\left(y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right)\right) - H_s\left(y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right)\right)\right) \\
&\quad + \lambda_1\left\|\widetilde{\beta}^*\right\|_1 + \frac{\lambda_2\left\|\widetilde{\beta}^*\right\|_2^2}{2} \\
&\geq C\sum_{i=1}^{l} H_1\left(y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right) \\
&\quad + C^*\sum_{i=l+1}^{l+2u}\left(H_1\left(y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right) - H_s\left(y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right)\right) \\
&\quad + \lambda_1\left\|\widetilde{\beta}\right\|_1 + \frac{\lambda_2\left\|\widetilde{\beta}\right\|_2^2}{2}.
\end{aligned}
\tag{A.2}
$$

It is simple to verify that both the loss function $H_1(z)$ and $H_s(z)$ are Lipschitz continuous, so we have

$$
\begin{aligned}
&C\sum_{i=1}^{l} H_1\left(y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right)\right) - C\sum_{i=1}^{l} H_1\left(y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right) \\
&\quad \leq C\sum_{i=1}^{l}\left|H_1\left(y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right)\right) - H_1\left(y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right)\right| \\
&\quad \leq M\sum_{i=1}^{l}\left|y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right) - y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right| \\
&\quad = \frac{M}{2}\left|\widetilde{\beta}_j - \widetilde{\beta}_l\right|\sum_{i=1}^{l}\left|\left(x_{ij} - x_{ik}\right)\right|.
\end{aligned}
\tag{A.3}
$$

Similarly, we can get

$$
\begin{aligned}
&C^*\sum_{i=l+1}^{l+2u} H_1\left(y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right)\right) - C^*\sum_{i=l+1}^{l+2u} H_1\left(y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right) \\
&\qquad \leq \frac{M_1}{2}\left|\widetilde{\beta}_j - \widetilde{\beta}_k\right|\sum_{i=l+1}^{l+2u}\left|\left(x_{ij} - x_{ik}\right)\right|, \\
&C^*\sum_{i=l+1}^{l+2u} H_s\left(y_i\left(x_i\widetilde{\beta} + \widetilde{\beta}_0\right)\right) - C^*\sum_{i=l+1}^{l+2u} H_s\left(y_i\left(x_i\widetilde{\beta}^* + \widetilde{\beta}_0^*\right)\right) \\
&\qquad \leq \frac{M_2}{2}\left|\widetilde{\beta}_j - \widetilde{\beta}_k\right|\sum_{i=l+1}^{l+2u}\left|\left(x_{ij} - x_{ik}\right)\right|,
\end{aligned}
\tag{A.4}
$$

where M_1 and M_2 are positive constants. As described in [11], we get $\|\widetilde{\beta}^*\|_1 - \|\widetilde{\beta}\|_1 \leq 0$ and $\|\widetilde{\beta}^*\|_2^2 - \|\widetilde{\beta}\|_2^2 = -(1/2)|\widetilde{\beta}_j - \widetilde{\beta}_k|$. Then combining (A.3), (A.4), and (A.2) implies that

$$
\begin{aligned}
&\frac{M}{2}\left|\widetilde{\beta}_j - \widetilde{\beta}_k\right|\sum_{i=1}^{l}\left|\left(x_{ij} - x_{ik}\right)\right| + \frac{M_1 + M_2}{2}\left|\widetilde{\beta}_j - \widetilde{\beta}_k\right| \\
&\quad \times \sum_{i=l+1}^{l+2u}\left|\left(x_{ij} - x_{ik}\right)\right| - \frac{\lambda_2}{4}\left|\widetilde{\beta}_j - \widetilde{\beta}_k\right|^2 \geq 0.
\end{aligned}
\tag{A.5}
$$

Let $M' = M_1 + M_2$, and (7) is obtained. For (8), we simply use the two inequalities $\sum_{i=1}^{l}|x_{ij} - x_{ik}| \leq \sqrt{l}\sqrt{2(1-\rho_l)}$, and $\sum_{i=l+1}^{l+2u}|x_{ij} - x_{ik}| \leq \sqrt{2u}\sqrt{2(1-\rho_u)}$. □

Acknowledgments

This work was supported by the National Key Basic Research Program of China (973 Program) under Grant no. 613148 and the National Science and Technology Major Project of the Ministry of Science and Technology under Grant no. 2010ZX03006-002, 2011ZX03005-003-03. The authors are grateful to the anonymous reviewers for their helpful comments.

References

[1] Y. Hong, S. Kwong, Y. Chang, and Q. Ren, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," *Pattern Recognition*, vol. 41, no. 9, pp. 2742–2756, 2008.

[2] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[3] J. Lin, J. Ming, and D. Crookes, "Robust face recognition with partial occlusion, illumination variation and limited training data by optimal feature selection," *IET Computer Vision*, vol. 5, no. 1, pp. 23–32, 2011.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, Hoboken, NJ, USA, 2nd edition, 2006.

[5] L. Talavera, C. Nord, and J. Girona, "Dependency-Based Feature Selection for Clustering Symbolic Data," *Intelligent Data Analysis*, vol. 4, no. 1, pp. 19–28, 2000.

[6] H. Elghazel and A. Aussem, "Feature selection for unsupervised learning using random cluster ensembles," in *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM '10)*, pp. 168–175, Sydney, Australia, December 2010.

[7] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, Cambridge, Mass, USA, 2006.

[8] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese, "Semi-supervised local Fisher discriminant analysis for dimensionality reduction," *Machine Learning*, vol. 78, no. 1-2, pp. 35–61, 2010.

[9] F. Bellal, H. Elghazel, and A. Aussem, "A semi-supervised feature ranking method with ensemble learning," *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1426–1432, 2012.

[10] Z. Zhao and H. Lu, "Semi-supervised feature selection via spectral analysis," in *Proceedings of the 7th SIAM International Conference on Data Mining*, pp. 641–646, April 2007.

[11] L. Wang, J. Zhu, and H. Zou, "The doubly regularized support vector machine," *Statistica Sinica*, vol. 16, no. 2, pp. 589–615, 2006.

[12] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society B*, vol. 67, no. 2, pp. 301–320, 2005.

[13] Y. Guibo, C. Yifei, and X. Xiaohui, "Efficient variable selection in support vector machines via the alternating direction method of multipliers," *Journal of Machine Learning Research*, vol. 15, pp. 832–840, 2011.

[14] T. Joachims, "Transductive inference for text classification using support vector machines," *Proceedings of the 16th International Conference on Machine Learning (ICML '99)*, pp. 200–209, 1999.

[15] O. Chapelle, V. Sindhwani, and S. S. Keerthi, "Optimization techniques for semi-supervised support vector machines," *Journal of Machine Learning Research*, vol. 9, pp. 203–233, 2008.

[16] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*, Oxford University Press, New York, NY, USA, 1995.

[17] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *Journal of Machine Learning Research (JMLR)*, vol. 7, pp. 1687–1712, 2006.

[18] R. T. Rockafellar, "A dual approach to solving nonlinear programming problems by unconstrained optimization," *Mathematical Programming*, vol. 5, pp. 354–373, 1973.

[19] C. Wu and X.-C. Tai, "Augmented lagrangian method, dual methods, and split bregman iteration for ROF, Vectorial TV, and high order models," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 300–339, 2010.

[20] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[21] Y. Saad, *Iterative Methods For Sparse Linear Systems*, Society for Industrial Mathematics, 2003.

[22] G.-B. Ye and X. Xie, "Split Bregman method for large scale fused Lasso," *Computational Statistics and Data Analysis*, vol. 55, no. 4, pp. 1552–1569, 2011.

[23] J. B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms*, Springer, Berlin, Germany, 1993.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.