

A New SVM Model for Classifying Genetic Data

Wang-Juh Chen, Hongbin Guo, Rosemary A Renaut
Arizona State University
{wangjuhchen, hb_guo, renaut}@asu.edu

Kewei Chen
The Banner Alzheimer's Institute
Kewei.Chen@bannerhealth.com

Abstract

We propose a new formulation of the Support Vector Machine (SVM) for classifying genetic data. It is based on the development of ideas from the method of total least squares, in which assumed error in measured data are incorporated in the model design. For genetic data the number of features is always far greater than the sample size. Consequently, in our method, we introduce Lagrange multipliers and solve for the dual variables. Instead of finding the minimum value of the Lagrangian function, we solve the nonlinear system of equations obtained from the Karush-Kuhn-Tucker conditions. We also implement complementarity constraints and incorporate weighting of the linear system by the inverse covariance matrix of the measured data. The proposed algorithm gives improved results and higher sensitivity for classifying a set of Alzheimer's Disease Positron Emission Tomography images as compared with SVM. It is also more robust to noise than SVM.

1. Introduction

There has been recently increased interest in algorithms based on the Support Vector Machine (SVM), [17], as providing efficient data mining classification. They have been found to be useful in many applications including pattern recognition, image, and text categorization [2], etc [18]. In the analysis of genome function, microarray data, which are obtained as a gene expression matrix, may be analysed using the SVM, [10, 3, 6, 16]. Chen et al, [5], propose a genetic fuzzy classification fusion model to combine multiple SVM classifiers. The experimental results obtained when applying this multiple SVM classifier to biomedical data demonstrate that it is more robust and reliable than using individual SVMs for classification. Medical images, such as Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET), may also be classified directly by treating the voxel data as features, or indirectly by using a regional feature extraction method [8].

In this paper we first present a brief overview of standard

formulations for the SVM. Many references exist which provide more details and an overview of recent developments, [11, 4, 18, 14, 1]. Then we remodel the soft margin SVM by an **errors-in-features** model [13]. The basic premise of the new model is to recognize that feature data are prone to error in the measurements, and thus errors can be treated accordingly. Dependent on whether the constraints, that define the classification of each data pair (feature space and its class), are imposed as equality or inequality constraints, different models result. Here we focus on the constraints imposed as inequalities, which introduces a Karush-Kuhn-Tucker (KKT) system and yields an algorithm in which we need to solve a nonlinear system of equations.

In Section 2 we present the existing and newly developed formulations. A numerical algorithm is developed in Section 3. The proposed algorithm is validated in Section 4 with conclusions in Section 5.

2. Methods

Given a training set $\{\mathbf{x}_m, t_m\}_{m=1}^M$ with input data $\mathbf{x}_m \in R^N$ and class labels $t_m \in \{-1, +1\}$, the purpose of the SVM is to find a decision function (classifier)

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b, \quad (1)$$

which separates the data into two classes, according to the sign of f . The hyperplane, given by $f(\mathbf{x}) = 0$, separates the two classes of data, assuming that they are separable, and is determined by maximizing the margin $2/\|\mathbf{w}\|_2^2$ between the two sets of points lying on the planes given by $f(\mathbf{x}) = \pm 1$, [17]. Function $\phi(\mathbf{x})$ can be a suitably chosen nonlinear function which maps the input space into a higher-dimensional space, although here we will primarily focus on the linear SVM in which $\phi(\mathbf{x}) = \mathbf{x}$. In this case, the classification of the training data provides the linear constraint equations

$$t_m[\mathbf{w}^T \phi(\mathbf{x}_m) + b] = t_m[\mathbf{w}^T \mathbf{x}_m + b] \geq 1.$$

Dependent on how the constraints for the training data are imposed, whether the data are separable or not, different

SVMs can be derived. We first review the standard approach and then consider an extension in which measurement errors are considered.

2.1. Hard Margin SVM

Suppose that we seek to find the optimal separating hyperplane by solving the following optimization problem

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_2^2, \text{ subject to} \\ t_m [\mathbf{w}^T \phi(\mathbf{x}_m) + b] \geq 1, \quad m = 1, \dots, M. \quad (2)$$

We introduce the matrices

$$A(m, :) = T(m, m) (\phi^T(\mathbf{x}_m)), \quad T = \text{diag}(t_1, t_2, \dots, t_m), \\ \text{where } A(m, :) \text{ is the } m^{\text{th}} \text{ row of } A \in \mathcal{R}^{M \times N},$$

and the vectors of length M

$$\mathbf{e} = [1, \dots, 1]^T, \quad \boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_M]^T.$$

Then the hard Margin SVM solves the following dual problem, which is a quadratic programming (QP) problem:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\text{argmin}} \frac{1}{2} \|A^T \boldsymbol{\alpha}\|_2^2 - \mathbf{e}^T \boldsymbol{\alpha}, \\ \text{subject to } \boldsymbol{\alpha}^T \mathbf{t} = 0 \text{ and } \boldsymbol{\alpha} \geq 0. \quad (3)$$

The positive entries of $\boldsymbol{\alpha}$ correspond to constraints in (2) which are *active* and are chosen as the **support vectors** for the classification. Zero Lagrange multipliers, $\alpha_m = 0$, correspond to the *inactive* constraints, those which are only satisfied as inequalities. The primal variables are obtained from KKT conditions

$$\mathbf{w} = A^T \boldsymbol{\alpha}, \quad (4)$$

with, generally, the intercept given as the average over the nonzero support vectors,

$$b = \text{av}(t_m - \mathbf{w}^T \phi(\mathbf{x}_m)), \quad (5)$$

where $\text{av}(\mathbf{x})$ for vector \mathbf{x} is the average of its components taken only over values for which $\alpha_m > 0$. Future data may be classified using (1), although this classification is typically written in terms of the dual variables

$$f(\mathbf{x}) = \sum_{m=1}^M \alpha_m t_m \phi(\mathbf{x}_m)^T \phi(\mathbf{x}) + b. \quad (6)$$

It is immediate from the definition of the matrix A that

$$\|A^T \boldsymbol{\alpha}\|_2^2 = \sum_{n=1}^N \sum_{m=1}^M t_n t_m \alpha_n \alpha_m \phi(\mathbf{x}_m)^T \phi(\mathbf{x}_n).$$

Thus introducing the kernel function $k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_m)^T \phi(\mathbf{x}_n)$, it is immediate to see that objective function (3) can be written entirely in terms of the kernel function. The kernel values need only be calculated initially hence avoiding the need to store the feature vectors $\phi(\mathbf{x}_m)$. When we use the linear map $\phi(\mathbf{x}_m) = \mathbf{x}_m$ the kernel is referred to as the dot product. Indeed the use of the kernel, commonly called the *kernel trick*, is one of the major advantages of the SVM, particularly for its inherent ability to reduce the dimension of the problem and thus to reduce computational complexity and cost.

2.2. Soft Margin SVM

In general the data are not linearly separable and there is no feasible solution to the Hard Margin SVM. Slack variables ξ introduced in the constraints (2) measure the classification error which leads to the Soft Margin SVM

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|_2^2 + \lambda \sum_{m=1}^M \|\xi_m\|_p, \quad (7)$$

$$\text{subject to } A\mathbf{w} + b\mathbf{t} \geq \mathbf{e} - \boldsymbol{\xi} \text{ and } \boldsymbol{\xi} \geq 0.$$

Here λ is a real positive number which trades off between the size of the margin and the total classification error. When $p = 1(2)$ we obtain the L1(2) Soft Margin SVM, resp, either of which lead to a convex QP problem [1, 4].

As for the Soft Margin SVM, the solution of (7) is found by first forming its Lagrangian, and then solving the KKT conditions for the primal variables, yielding a maximization for the dual problem. The Lagrangian for (7) is

$$L_{\text{PS}}(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{w}\|_2^2 + \lambda \mathbf{e}^T \boldsymbol{\xi} \\ - \boldsymbol{\alpha}^T [A\mathbf{w} + b\mathbf{t} + \boldsymbol{\xi} - \mathbf{e}] - \boldsymbol{\beta}^T \boldsymbol{\xi},$$

where $\boldsymbol{\beta} \geq 0$ is a new set of Lagrange multipliers introduced to enforce the positivity of slack variables $\boldsymbol{\xi}$. For this Lagrangian the KKT conditions are augmented by the equations

$$\boldsymbol{\alpha} + \boldsymbol{\beta} = \lambda \mathbf{e}, \quad \boldsymbol{\beta} \geq 0, \quad \boldsymbol{\xi} \geq 0, \quad \beta_m \xi_m = 0, \quad m = 1, \dots, M,$$

and the constraints are modified to

$$\alpha_m [A\mathbf{w} + b\mathbf{t} + \boldsymbol{\xi} - \mathbf{e}]_m = 0, \quad m = 1, \dots, M.$$

Thus the Soft Margin SVM is obtained by solving the convex QP (3), again the kernel is used, with additional upper bound constraints imposed on the support vectors, $0 \leq \alpha \leq \lambda$. Data are still classified by (6) using (4) and (5) for calculating \mathbf{w} and the intercept.

2.3. Development of the Errors in Features Model

Notice now that matrix $A = T\Phi^T$, where Φ is the mapped feature matrix. In the soft formulation of the SVM some misclassification is permitted. Indeed the soft SVM can be reformulated in terms of a *loss + penalty* functional

$$\min_{\mathbf{w}, b} \sum_{m=1}^M [1 - t_m f(\mathbf{x}_m)]_+ + \frac{\mu}{2} \|\mathbf{w}\|_2^2,$$

where $(\eta)_+ = \eta$ if $\eta > 0$ otherwise $(\eta)_+ = 0$, and $\mu = 1/\lambda$. This shows specifically that the slack variables ξ account in (7) for the potential misclassification error, perhaps due to the evaluation of Φ . Now consider the case in which $\Phi = X$, i.e. the linear mapping. Then instead of introducing slack variables, we can account directly for error in the feature space by a matrix E , yielding the linear SVM with error in data modeled as follows

$$\begin{aligned} \min_{\mathbf{w}, b, E} \quad & \frac{1}{2} \|E\|_F^2 + \frac{\mu}{2} \|\mathbf{w}\|_2^2, \\ \text{subject to} \quad & (A + E)\mathbf{w} + b\mathbf{t} \geq \mathbf{e}. \end{aligned} \quad (8)$$

Here $\|\cdot\|_F$ denotes the Frobenius norm. If we replace the inequality constraints by equalities, the formulation would yield a ‘‘so-called’’ regularized mixed LS-TLS problem, see [13]. Instead of the slack variables accounting for misclassification in (7), the errors-in-features SVM (EFSVM) very specifically measures the total feature space error.

The Lagrangian for (8) is given by

$$\begin{aligned} L_{\text{EF}}(\mathbf{w}, b, E, \boldsymbol{\alpha}) = & \frac{1}{2} \|E\|_F^2 + \frac{\mu}{2} \|\mathbf{w}\|_2^2 \\ & - \boldsymbol{\alpha}^T [(A + E)\mathbf{w} - \mathbf{e} + b\mathbf{t}], \quad \boldsymbol{\alpha} \geq 0. \end{aligned} \quad (9)$$

The KKT conditions are

$$\begin{aligned} \nabla_E L_{\text{EF}} = E - \boldsymbol{\alpha}\mathbf{w}^T &= 0, \\ \nabla_{\mathbf{w}} L_{\text{EF}} = \mu\mathbf{w} - (A + E)^T \boldsymbol{\alpha} &= 0, \\ \frac{\partial L_{\text{EF}}}{\partial b} = \boldsymbol{\alpha}^T \mathbf{t} &= 0, \end{aligned} \quad (10)$$

$$\begin{aligned} D_{\boldsymbol{\alpha}}((A + E)\mathbf{w} - \mathbf{e} + b\mathbf{t}) &= 0, \\ (A + E)\mathbf{w} - \mathbf{e} + b\mathbf{t} &\geq 0, \\ \boldsymbol{\alpha} \geq 0, \text{ where } D_{\boldsymbol{\alpha}} = \text{diag}(\boldsymbol{\alpha}). \end{aligned} \quad (11)$$

These yield immediately $E = \boldsymbol{\alpha}\mathbf{w}^T$, $\mathbf{w} = \mu^{-1}(A + E)^T \boldsymbol{\alpha}$ and for $\|\boldsymbol{\alpha}\|_2^2 \neq \mu$,

$$\begin{aligned} \mathbf{w} &= \frac{1}{\mu - \|\boldsymbol{\alpha}\|_2^2} A^T \boldsymbol{\alpha}, \\ E &= \frac{1}{\mu - \|\boldsymbol{\alpha}\|_2^2} \boldsymbol{\alpha}\boldsymbol{\alpha}^T A, \\ \boldsymbol{\alpha}^T \mathbf{t} &= 0. \end{aligned}$$

Notice here the expression for E , for which the derivation is provided in the Appendix, replaces the more computationally expensive expression $(\mu I - \boldsymbol{\alpha}\boldsymbol{\alpha}^T)^{-1} \boldsymbol{\alpha}\boldsymbol{\alpha}^T A$. The derivation is provided in the Appendix. Substituting for E and \mathbf{w} in (10) and (11) we see that an approach to find $\boldsymbol{\alpha}$ is to solve the nonlinear system of equations given by

$$\begin{aligned} \mathbf{h}(\boldsymbol{\alpha}, b) = \begin{pmatrix} \boldsymbol{\alpha}^T \mathbf{t} \\ D_{\boldsymbol{\alpha}} \mathbf{g}(\boldsymbol{\alpha}, b) \end{pmatrix} &= 0, \\ \text{in the region } \boldsymbol{\alpha} \geq 0, \mathbf{g}(\boldsymbol{\alpha}, b) \geq 0, & \text{ where} \\ \mathbf{g}(\boldsymbol{\alpha}, b) = qAA^T \boldsymbol{\alpha} + \boldsymbol{\alpha}^T AA^T \boldsymbol{\alpha} \cdot \boldsymbol{\alpha} + q^2(b\mathbf{t} - \mathbf{e}) & \\ \text{and } q(\boldsymbol{\alpha}) = \mu - \|\boldsymbol{\alpha}\|_2^2 \neq 0. & \end{aligned} \quad (12)$$

This is a nonlinear system with $M + 1$ equations and variables and can be further reduced by introducing the complementarity condition. Specifically, from (12), we know that either $\boldsymbol{\alpha}$ or \mathbf{g} should be zero, and because of the nonnegativity conditions, (12) can be reduced to

$$\begin{cases} \boldsymbol{\alpha}^T \mathbf{t} = 0 \\ \boldsymbol{\alpha}^T \mathbf{g} = 0 \end{cases} \text{ in the region } \boldsymbol{\alpha} \geq 0, \mathbf{g} \geq 0,$$

where

$$\boldsymbol{\alpha}^T \mathbf{g} = q \cdot \boldsymbol{\alpha}^T AA^T \boldsymbol{\alpha} + \boldsymbol{\alpha}^T AA^T \boldsymbol{\alpha} \cdot \boldsymbol{\alpha}^T \boldsymbol{\alpha} + q^2(b\boldsymbol{\alpha}^T \mathbf{t} - \boldsymbol{\alpha}^T \mathbf{e}).$$

Because $\boldsymbol{\alpha}^T AA^T \boldsymbol{\alpha} = \|A^T \boldsymbol{\alpha}\|_2^2$, $q = \mu - \|\boldsymbol{\alpha}\|_2^2$ and $\boldsymbol{\alpha}^T \mathbf{t} = 0$,

$$\begin{aligned} \boldsymbol{\alpha}^T \mathbf{g} &= q \|A^T \boldsymbol{\alpha}\|_2^2 + \|A^T \boldsymbol{\alpha}\|_2^2 \cdot \|\boldsymbol{\alpha}\|_2^2 - q^2 \boldsymbol{\alpha}^T \mathbf{e} \\ &= \|A^T \boldsymbol{\alpha}\|_2^2 (q + \|\boldsymbol{\alpha}\|_2^2) - q^2 \boldsymbol{\alpha}^T \mathbf{e} \\ &= \|A^T \boldsymbol{\alpha}\|_2^2 \cdot \mu - q^2 \boldsymbol{\alpha}^T \mathbf{e}. \end{aligned}$$

Thus solving (12) is equivalent to finding the solution of

$$\begin{aligned} \mathbf{F}(\boldsymbol{\alpha}, b) = \begin{cases} \boldsymbol{\alpha}^T \mathbf{t} = 0 \\ \|A^T \boldsymbol{\alpha}\|_2^2 \cdot \mu - q^2 \boldsymbol{\alpha}^T \mathbf{e} = 0 \end{cases} & \quad (13) \\ \text{in the region } \boldsymbol{\alpha} \geq 0, \mathbf{g}(\boldsymbol{\alpha}, b) \geq 0. & \end{aligned}$$

This is a typical nonlinear system of equations and can be solved by using Newton’s method. We discuss the details of the chosen Newton algorithm in the next section, but first we note that other approaches might have been chosen for finding $\boldsymbol{\alpha}$. For example, substituting KKT conditions into (9) would lead to a difficult nonlinear optimization problem. On the other hand, instead of seeking to solve the nonlinear system of equations directly one might solve the equivalent nonlinear least squares minimization using a Gauss-Newton (GN) method. Our experience with the GN algorithm for this problem suggests that difficulties arise due to severe ill-conditioning of the Jacobian. We therefore present our approach for the direct solution of the nonlinear equations by Newton’s method, with linearization of $\mathbf{h}(\boldsymbol{\alpha}, b)$.

3. The numerical algorithm

In computing the Newton step for the solution of (13) we linearize $\mathbf{h}(\boldsymbol{\alpha}, b)$ and obtain the following linear system

$$\begin{pmatrix} \text{diag}(\mathbf{g}) + D_{\boldsymbol{\alpha}}G(\boldsymbol{\alpha}) & q^2 D_{\boldsymbol{\alpha}}\mathbf{t} \\ \mathbf{t}^T & 0 \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\alpha} \\ \Delta b \end{pmatrix} = \begin{pmatrix} -D_{\boldsymbol{\alpha}}\mathbf{g} \\ -\boldsymbol{\alpha}^T\mathbf{t} \end{pmatrix}, \quad (14)$$

where $G(\boldsymbol{\alpha}) = qAA^T - 2AA^T\boldsymbol{\alpha}\boldsymbol{\alpha}^T + \boldsymbol{\alpha}^TAA^T\boldsymbol{\alpha} \cdot I + 2\boldsymbol{\alpha}\boldsymbol{\alpha}^TAA^T + 4q\mathbf{e}\boldsymbol{\alpha}^T$, and use a line search update

$$\begin{pmatrix} \boldsymbol{\alpha}^{(k+1)} \\ b^{(k+1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha}^{(k)} \\ b^{(k)} \end{pmatrix} + \omega \begin{pmatrix} \Delta\boldsymbol{\alpha} \\ \Delta b \end{pmatrix}.$$

The line search parameter $\omega \in [0, 1]$ is used to adjust the step length such that the new solution decreases sufficiently according to the Armijo rule

$$\|\mathbf{F}(\boldsymbol{\alpha} + \omega\Delta\boldsymbol{\alpha})\| < (1 - \tau\omega)\|\mathbf{F}(\boldsymbol{\alpha})\|,$$

where $\tau \in [0, 1]$. Here we use $\tau = 10^{-4}$ as suggested in Kelly [15].

3.1. Regularization

In solving the nonlinear system of equations (13) by Newton's method, we realize that the linear system (14) is always ill-posed and can not be solved directly. In order to deal with the ill-posedness, we introduce, as in the Levenberg-Marquardt algorithm for nonlinear least squares, a regularization term. Specifically, suppose we have an ill-posed system $A\mathbf{x} = \mathbf{b}$ and want to find the solution, after introducing the regularization parameter $\lambda > 0$, the system becomes $(A^T A + \lambda I)\mathbf{x} = A^T \mathbf{b}$ [7] and the solution is given by

$$\begin{aligned} \mathbf{x}^* &= \underset{\mathbf{x}}{\text{argmin}} \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \\ &= (A^T A + \lambda I)^{-1} A^T \mathbf{b}. \end{aligned}$$

The parameter λ is changed at each Newton iteration. In our implementation we find that the iterative Tikhonov regularization initialized with $\lambda^{(0)} = 10^{-5}\sigma_0^{(0)}$, where $\sigma_0^{(0)}$ is the largest singular value of the initial matrix A , and subsequent values decreased according to $\lambda^{(k)} = \frac{\lambda^{(k-1)}}{2}$ is suitable.

3.2. Initial point

The initial point must be chosen in order to satisfy the constraints in (13). To generate a nonnegative vector $\boldsymbol{\alpha}^{(0)}$ is an easy task, but to make sure $\mathbf{g}(\boldsymbol{\alpha}^{(0)}, b^{(0)}) \geq 0$ is more subtle. Nonnegative $\boldsymbol{\alpha}^{(0)}$ can only guarantee that the second term of $\mathbf{g}(\boldsymbol{\alpha}^{(0)}, b^{(0)})$, $\boldsymbol{\alpha}^T A A^T \boldsymbol{\alpha} \cdot \boldsymbol{\alpha}$, is positive. By setting $q(\boldsymbol{\alpha}^{(0)})$ to be a small number, i.e. depends

on the data scale, and $b^{(0)} = 0$, the third term is a negative vector smaller than the sum of the previous two, and this yields a valid initial point for the system. It follows that $\mu \approx \|\boldsymbol{\alpha}^{(0)}\|_2^2$, so a suitable choice is the normalization $\boldsymbol{\alpha}^{(0)} = \sqrt{\mu}|\boldsymbol{\alpha}^{(0)}|/\|\boldsymbol{\alpha}^{(0)}\|_2$.

3.3. Covariance matrix

We first suppose that all the features are equally significant, so the solution can be found as addressed so far. But, especially for biological data sets, different genes (features), affect specific pathways at different levels, and some are totally irrelevant. Weighting the data by the inverse variance for each feature places least weight on data with very high variance. After introducing the feature covariance matrix F , generally $F = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \dots, \sigma_N^2)$ and σ_i^2 the variance in feature i , the nonlinear system of equations is exactly the same as in (13) except we replace A everywhere by $\tilde{A} = AF^{-\frac{1}{2}}$. On the other hand, features with very small variance are less likely to contribute to the classification. We thus prune the data in advance by removing features with low variance, eg variance $< 10^{-3}$. This choice is problem dependent. We note also that greater impact may be achieved by forming the complete covariance matrix and then using its Cholesky factorization to find weighting matrix $F^{\frac{1}{2}} = L$, where $F = LL^T$ is the Cholesky factorization. This is a topic for future research.

3.4. Pseudo Code

The overall algorithm is summarized below

Outline for EFSVM algorithm

Set $\boldsymbol{\alpha}^{(0)}, b^{(0)}$ and check for $\boldsymbol{\alpha}^{(0)} \geq 0, \mathbf{g}^{(0)} \geq 0$

Do until convergence or the max step exceeded

Solve $\nabla \mathbf{h} \cdot \Delta(\boldsymbol{\alpha}, b) = -\mathbf{h}$

Find step length ω s.t. $\boldsymbol{\alpha}^{(k+1)} > 0, \mathbf{g}^{(k+1)} > 0$

and update $\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \omega\Delta\boldsymbol{\alpha}$

and $b^{(k+1)} = b^{(k)} + \omega\Delta b$

End

4. Data Sets and Experiments

4.1. Data Set

We test our method using Positron Emission Tomography (PET) data for Alzheimer's disease classification.

The PET image data sets are obtained from the ADNI [19] website and preprocessed (normalization, registration, etc.) at Banner Alzheimer's Institute. The dot product kernel function is generated using the Statistical Parametric Mapping (SPM) [9] software. 97 subjects are grouped into Alzheimer's disease (AD) patients (44 samples) and Normal Controls (53 samples). We also use 159 subjects from

a data set of subjects with mild cognitive impairment (MCI) which is a transition state from NC to AD. For this data set we demonstrate our ability to distinguish from among the MCI subjects those that convert from NC to AD. Such subjects are denoted as *converters*. All PET images were taken at baseline (month 0) and month 12.

4.2. Experimental Setup

In order to validate the EFSVM, we compare our results with a standard SVM (MathWorks) using leave one out cross validation. To choose a $\mu > 0$ to trade off between the size of the margin and the error in feature space, $\mu = 10$ is used on all the data sets in this paper.

In order to assess the impact of error in measurements, we perturbed the data sets by different levels of noise. Given a data set X , the new data set \tilde{X} with perturbation, i.e. ϵ , will be $\tilde{X} = (1 + \epsilon \cdot \text{randn})X$ where randn is a matrix with values drawn from a normal distribution with mean zero and unit standard deviation.

For data which can be classified into only two groups, the results of any experimental simulation with known results yields the classification of each result as a True Positive (TP), True Negative (TN), False Positive (FP), or False Negative (FN). An algorithm's performance can thus be assessed in terms of its accuracy, specificity, and sensitivity, which are defined as follows: Accuracy = $\frac{TP+TN}{N}$, where N is the total number of examples, Specificity = $\frac{TN}{TN+FP}$, and Sensitivity = $\frac{TP}{TP+FN}$. The accuracy alone is insufficient for assessing algorithm performance with respect to diagnostic situations. Rather sensitivity is a very important indicator because it measures the ability of an algorithm to recognize patients with disease.

4.3. Results and Discussion

Figure 1. Accuracy on different levels of perturbation

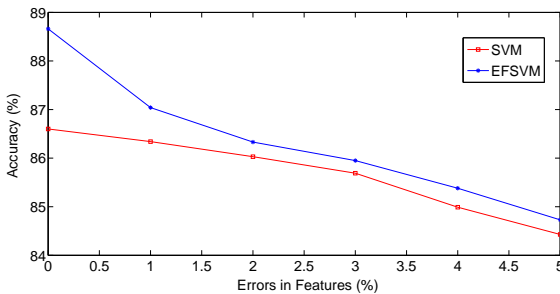


Table 1 shows the accuracy, sensitivity, and specificity for classifying PET image data using leave one out cross

Table 3. The accuracy with different levels of perturbation

	SVM	EFSVM
0%	86.60(0)	88.66(0)
1%	86.34(0.91)	87.04(0.83)
2%	86.03(1.39)	86.33(1.52)
3%	85.69(2.02)	85.95(2.22)
4%	84.99(2.67)	85.38(2.91)
5%	84.43(3.11)	84.73(3.57)

validation. Having the same specificity as SVM, EFSVM exhibits the same ability to determine a healthy person. But with higher sensitivity, EFSVM demonstrates more confidence in predicting an AD patient. This is very important for disease screening. We also looked at the misclassified groups (False Positive/Negative) from both methods and found that the misclassifications by EFSVM are a subset of those by SVM. This means that EFSVM not only preserves the results from SVM but is also better in some cases.

To demonstrate the classification ability of EFSVM for data sets with some errors in features, we examined perturbed data sets for the PET image data with ϵ from 1% to 5% as described in Section 4.2. Table 3 gives the results for both SVMs on the PET image data at baseline with the mean and standard deviation (in parentheses) of the accuracy over 100 runs with leave one out cross validation. From Figure 1 and Table 3, we can see that both methods yield lower accuracy as ϵ increases, but EFSVM always outperforms SVM. For example, although EFSVM accuracy drops 1.62% when $\epsilon = 1\%$, the accuracy is still better than the SVM accuracy when $\epsilon = 0$. This indicates that EFSVM confirms its design strategy with regard to being more robust to error than SVM.

Furthermore, in order to show the ability to distinguish the converters from nonconverters within the MCI group, we use the classifier which is trained for distinguishing AD and NC subjects to classify the MCI subjects. Not all MCI will be converters, actually, only 35 out of 159 in this data set. We should expect lower specificity in the result due to the unbalanced class distribution. Once again, from Table 2, we see that EFSVM yields a higher sensitivity than SVM and includes all the true positives (TP) of the SVM. This suggests that EFSVM has the potential to aid early diagnosis of AD.

The condition number for all systems, i.e. across all training sets, is between 10^7 and 10^{24} and the largest singular value for all systems is between 10^4 and 10^9 . Without regularization, the nonlinear system (14) can not be solved directly. Applying the initial regularization parameters, $\lambda^{(0)}$, chosen dependent on the largest singular value,

Table 1. The accuracy of EFSVM and SVM for AD data sets

Method	Month	Accuracy	TN	TP	FN	FP	Specificity	Sensitivity
EFSVM	0	88.66(86/97)	50	36	8	3	94.3	81.8
SVM	0	86.60(84/97)	50	34	10	3	94.3	77.3
EFSVM	12	89.70(87/97)	51	36	8	2	96.2	81.8
SVM	12	88.66(86/97)	51	35	9	2	96.2	79.5

Table 2. The accuracy of EFSVM and SVM for MCI data sets at month 12

	MCI classified as		Statistics					
	AD(converters)	NC(converters)	TN	TP	FN	FP	Specificity	Sensitivity
EFSVM	84(28)	75(7)	68	28	7	56	54.84	80.00
SPM SVM	81(26)	78(9)	69	26	9	55	55.65	74.29

between 1 and 10^4 , the algorithm converges within 26 to 32 steps so that the last $\lambda^{(k)}$ is between 10^{-6} to 10^{-8} . This shows that the regularization not only makes the problem solvable, but also allows the algorithm to converge in fewer steps. It is possible to solve the system using $\lambda^{(k)} = \lambda$ fixed for all k but, results not reported here, confirm that the algorithm's convergence is far more sensitive to the correct choice of the fixed parameter λ , which is data set dependent.

5. Conclusions

In this paper we propose a new classification algorithm, the EFSVM, in which we modify the SVM to better model the errors-in-features. In particular, comparing to conventional SVM, EFSVM demonstrates higher sensitivity in predicting Alzheimer's disease and distinguishing MCI subjects from normal controls. This is of high interest in pre-clinical and clinical research for Alzheimer's disease.

6. Appendix

The Sherman-Morrison formula,

$$(A - \mathbf{u}\mathbf{w}^T)^{-1} = A^{-1} + A^{-1}\mathbf{u}(1 - \mathbf{w}^T A^{-1}\mathbf{u})^{-1}\mathbf{w}^T A^{-1},$$

where \mathbf{u} and \mathbf{w} are vectors, gives the inverse of a matrix resulting from a rank-one modification [12]. To prove that the two formulas for E in Section 2.3 are the same, we set $A = \mu I$, $\mathbf{u} = \boldsymbol{\alpha}$ and $\mathbf{w} = \boldsymbol{\alpha}$ to give

$$\begin{aligned} & (\mu I - \boldsymbol{\alpha}\boldsymbol{\alpha}^T)^{-1} \\ &= \mu^{-1}I + \mu^{-1}I\boldsymbol{\alpha}(1 - \boldsymbol{\alpha}^T\mu^{-1}I\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\mu^{-1}I \\ &= \mu^{-1}\left(I + \frac{\boldsymbol{\alpha}}{\mu - \boldsymbol{\alpha}^T\boldsymbol{\alpha}}\boldsymbol{\alpha}^T\right). \end{aligned}$$

Then

$$\begin{aligned} E &= (\mu I - \boldsymbol{\alpha}\boldsymbol{\alpha}^T)^{-1}\boldsymbol{\alpha}\boldsymbol{\alpha}^T A \\ &= \mu^{-1}\left(I + \frac{\boldsymbol{\alpha}}{\mu - \boldsymbol{\alpha}^T\boldsymbol{\alpha}}\boldsymbol{\alpha}^T\right)\boldsymbol{\alpha}\boldsymbol{\alpha}^T A \\ &= \mu^{-1}\left(\boldsymbol{\alpha} + \frac{\boldsymbol{\alpha}^T\boldsymbol{\alpha}}{\mu - \boldsymbol{\alpha}^T\boldsymbol{\alpha}}\boldsymbol{\alpha}\right)\boldsymbol{\alpha}^T A \\ &= \frac{1 - \mu^{-1}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \mu^{-1}\boldsymbol{\alpha}^T\boldsymbol{\alpha}}{\mu - \boldsymbol{\alpha}^T\boldsymbol{\alpha}}\boldsymbol{\alpha}\boldsymbol{\alpha}^T A \\ &= \frac{1}{\mu - \|\boldsymbol{\alpha}\|_2^2}\boldsymbol{\alpha}\boldsymbol{\alpha}^T A, \end{aligned}$$

as given in Section 2.3.

References

- [1] S. Abe. *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [2] J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic. Training text classifiers with SVM on very few positive examples. Technical report, Microsoft Research, April 2003.
- [3] M. P. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262–267, Jan 4 2000.
- [4] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.
- [5] X. Chen, Y. Li, R. Harrison, and Y.-Q. Zhang. Genetic fuzzy classification fusion of multiple svms for biomedical data. *J. Intell. Fuzzy Syst.*, 18(6):527–541, 2007.

- [6] M. L. Chow, E. J. Moler, and I. S. Mian. Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiological genomics*, 5(2):99–111, Mar 8 2001.
- [7] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Kluwer Academic Publishers, Dordrecht ; Boston, 1996.
- [8] Y. Fan, S. M. Resnick, and C. Davatzikos. Feature selection and classification of multiparametric medical images using bagging and SVM. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6914 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Apr. 2008.
- [9] W. T. C. for Neuroimaging. Statistical parametric mapping, December 2005.
- [10] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics (Oxford, England)*, 16(10):906–914, Oct 2000.
- [11] S. R. Gunn. Support vector machines for classification and regression. Technical report, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, May 1998.
- [12] M. T. Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill, 2002.
- [13] S. V. Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, Philadelphia, USA, 1991.
- [14] V. Kecman. *Support Vector Machines - An Introduction*, volume 177. Springer Berlin / Heidelberg, 2005.
- [15] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, USA, 2003.
- [16] S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J. P. Mesirov, and T. Poggio. Support vector machine classification of microarray data. In *AI Memo 1677, Massachusetts Institute of Technology*, 1999.
- [17] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [18] L. Wang. *Support Vector Machines: Theory and Applications (Studies in Fuzziness and Soft Computing)*. Springer, August 2005.
- [19] M. W. Weiner. Alzheimer’s disease neuroimaging initiative, 2008. <http://www.loni.ucla.edu/ADNI/>.