Published in final edited form as:
Stat Anal Data Min. 2011 December ; 4(6): 604–611. doi:10.1002/sam.10141.

A Novel Support Vector Classifier for Longitudinal High-dimensional Data and Its Application to Neuroimaging Data

Shuo Chen and **F. DuBois Bowman**
Department of Biostatistics and Bioinformatics, Emory University, Atlanta, GA 30322

Abstract

Recent technological advances have made it possible for many studies to collect high dimensional data (HDD) longitudinally, for example images collected during different scanning sessions. Such studies may yield temporal changes of selected features that, when incorporated with machine learning methods, are able to predict disease status or responses to a therapeutic treatment. Support vector machine (SVM) techniques are robust and effective tools well-suited for the classification and prediction of HDD. However, current SVM methods for HDD analysis typically consider cross-sectional data collected during one time period or session (e.g. baseline). We propose a novel support vector classifier (SVC) for longitudinal HDD that allows simultaneous estimation of the SVM separating hyperplane parameters and temporal trend parameters, which determine the optimal means to combine the longitudinal data for classification and prediction. Our approach is based on an augmented reproducing kernel function and uses quadratic programming for optimization. We demonstrate the use and potential advantages of our proposed methodology using a simulation study and a data example from the Alzheimer's disease Neuroimaging Initiative. The results indicate that our proposed method leverages the additional longitudinal information to achieve higher accuracy than methods using only cross-sectional data and methods that combine longitudinal data by naively expanding the feature space.

Keywords

fMRI; PET; prediction; classification; support vector classifier; Alzheimer's disease

1 Introduction

Current biomedical technology enables the collection of high-dimensional data (HDD) to gain insights regarding genomic, proteomic, and *in vivo* neural processing properties. Moreover, such HDD are more commonly being collected longitudinally, potentially revealing changes in biological properties that may provide clues to disease diagnosis, progression, or recovery. Machine learning tools have been widely applied for HDD classification and prediction (Mitchell *et al*., 2004; LaConte *et al*., 2005 Chen *et al*., 2007). Support vector machine methods are among the most popular machine learning techniques due to their high prediction accuracy and robustness (Vapnik, 1998; Mourao *et al*., 2005;Fu *et al*., 2008; Craddock *et al*., 2009). However, most current machine learning methods have been developed for cross-sectional rather than longitudinal high-dimensional data (LHDD) analysis. The "ideal" methodology for LHDD would take advantage of the additional data to determine temporal trends of features and use them as inputs within machine learning

models. However, in practice the temporal trends are usually unknown, and currently no such model exists for simultaneously determining the temporal trends and building the classification model.

To address classification or prediction objectives in context of LHDD, one may opt to use data from only a single time point, e.g. baseline data. Another potential approach for handling LHDD is a naive procedure of simply combining the longitudinal data as independent sources of information. Using data from only a single time point or using longitudinal data as independent sources of information may lead to substantial information loss and may not fully capitalize on the available data. One may also consider fitting preliminary models, for example, using logistic regression for each feature, and then using the resulting estimates to preset the temporal trends for classification. Since this approach uses classification outcome of interest in the preliminary modeling stage, presetting temporal parameters for each feature using model based estimates would lead to the vast danger of overfitting and pose difficulty for the following feature selection procedure.

In this paper, we propose a novel support vector classifier (SVC) for LHDD that extracts key features of each cross-sectional component as well as temporal trends between these components for the purpose of classification and prediction. The objective function of our new method incorporates two groups of estimands: the decision hyperplane function parameters and the temporal trend parameters that determine an optimal way to combine the longitudinal data. The objective function is derived from maximizing the margin width, with error-tolerated correct classification constraints. Within the framework of the Lagrange (Wolfe) dual of the objective function, we augment the dimension of the Hessian matrix by incorporating the temporal trend parameters. Then, we apply quadratic programming techniques to optimize the classification parameters and temporal trend parameters. With the kernels satisfying Mercer's conditions, the objective function is convex, leading to a finite dimensional representation of the decision function. The framework allows feature selection with unknown temporal trend parameters through recursive feature elimination (RFE) procedures.

Generally, our proposed framework is applicable to any type of high dimensional data that are measured longitudinally. For example, in the application to neuroimaging data, our method is applicable to longitudinal/multi-session studies collecting fMRI, PET, EEG, and MEG data. The longitudinal property refers to multiple scanning sessions (e.g. images or collections of images acquired on different days). Importantly, for some neuroimaging data (e.g. fMRI data), there may be a series of images measured at different time points within one session. Therefore, we usually use features reflecting various summaries from the original data at each session. For example, the features in our method may include functional connectivity, localized activity summary statistics (first level analysis results for fMRI data), or frequency domain summary statistics. Hence with appropriate summary statistics, our approach can handle a range of high-dimensional data modalities.

The rest of the paper is organized as follows. In Section 2, we present the new longitudinal SVC and provide an accompanying computational strategy. Furthermore, we discuss its extension to nonlinear kernels and RFE based feature selection algorithm. In Section 3, we

examine the classification performance of the proposed method for a data example and in a stimulation study. Section 4 concludes the paper with a summary and a discussion of the major strengths of our novel SVC for LHDD.

2 Methods

2.1 Classical Support Vector Classifier

SVC is a popular kernel machine learning algorithm that is derived to solve classification problems (Vapnik, 1996). For one subject indexed by *s*, the *p* dimensional feature space is denoted as $\boldsymbol{x}_s \in R^p$, for $s = 1, 2\ldots, N$ and group indicators $y_s \in \{-1, 1\}$ denote a binary state such as disease status (positive/negative) or treatment response (recovery or not). A classifier is defined by constructing a separating function (or hyperplane) $h(\boldsymbol{x}_s) = \boldsymbol{w} \cdot \boldsymbol{x}_s + b$ and then generating $\hat{y}_i = sign(h(\boldsymbol{x}_s))$, if the data are linearly separable. The SVC chooses the unique hyperplane that maximizes the margins, which are the distances between the hyperplane and the support vectors. For cases when data are not linearly separable, a 'soft margin' is introduced that allows some data points to be misclassified. Therefore, the SVC is subject to optimize the following objective function:

$$\min_{w} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{s=1}^{N}\xi_s \quad s=1,2,\ldots N, \quad (2.1)$$

subject to

$$y_s\left(\boldsymbol{w}\cdot\boldsymbol{x}_s + b\right) \geq 1-\xi_s, \quad and \quad \xi_s \geq 0.$$

where ξ_s is the distance of the subject *s* from its correct side of the margin and the constraint constant *C* is the tuning parameter regarding the tolerance level of misclassification.

Then, we obtain the Lagrange (Wolfe) dual by substituting $\boldsymbol{w} = \sum_{s=1}^{N} y_s \alpha_s \boldsymbol{x}_s$ to the Lagrange primal function of formula 2.1.

$$\min_{\alpha_s} \frac{1}{2}\sum_{s,s'} \alpha_s \alpha_{s'} y_s y_{s'} \left\langle \Phi\left(\boldsymbol{x}_s\right), \Phi\left(\boldsymbol{x}_{s'}\right)\right\rangle - \sum_{s=1}^{N}\alpha_s \quad for \quad s \quad and \quad s' = 1,2,\ldots N, \quad (2.2)$$

subject to

$$C \geq \alpha_s \geq 0, \quad and \quad \sum_{s} \alpha_s y_s = 0.$$

where $\mathbf{K}\left(\boldsymbol{x}_s, \boldsymbol{x}_{s'}\right) = \left\langle \Phi\left(\boldsymbol{x}_s\right), \Phi\left(\boldsymbol{x}_{s'}\right)\right\rangle$ means that we first map data into a higher dimension through the function Φ(.), then take the inner product of the mapped vectors. The formula 2.2 could be expressed as:

$$\min_{\alpha} \frac{1}{2}\alpha^T \mathbf{G}\alpha - 1'\alpha \quad (2.3)$$

where , **G** is a Gram matrix ($N \times N$) satisfying the Mercer's condition (requiring **G** is at least semi positive definite) multiplied by corresponding group labels, $\boldsymbol{\alpha}$ is a 1 by N vector of estimands. $G_{s,s'}$ is $\langle \Phi(x_s), \Phi(x_{s'}) \rangle y_s y_{s'}$. Then, the 'Wolfe' dual is well suited for quadratic programming (QP) optimization programs in most software. The objective function in formula (2.3) can be also considered as the sum of penalty and loss functions in terms of reproducing kernel Hilbert space with $h(\cdot) = \sum_{s=1}^{N} \alpha_s y_s \mathbf{K}(\boldsymbol{x}_s, \cdot) \in H_K$ (Wahba, 1990; Hastie and Tibshirani, 1990). Once the separating hyperplane has been determined through quadratic programming optimization, the class label of a new observation $\boldsymbol{x}_{new}$ can be determined by the sign function of $h(\boldsymbol{x}_{new}) = \sum_{s=1}^{N} \alpha_s y_s \mathbf{K}(\boldsymbol{x}_s, \boldsymbol{x}_{new}) + b$.

2.2 Longitudinal Support Vector Classifier - LSVC

Consider longitudinal data collected from N subjects at T measurement occasions or scanning sessions, with p features quantified during each session. The expanded feature matrix is then $T N$ by p. Let $\boldsymbol{x}_{s,t}$ be used to represent the features collected for one subject s at time t. Hence, our aim is to classify each individual $\boldsymbol{x}_s^{\sim} = \{\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, \ldots, \boldsymbol{x}_{s,T}\}'$ to a certain group $y_s \in \{-1, 1\}$. We characterize linear trends of change: $\boldsymbol{x}_s = \boldsymbol{x}_{s,1} + \beta_1 \boldsymbol{x}_{s,2} + \beta_2 \boldsymbol{x}_{s,3} \ldots + \beta_{T-1} \boldsymbol{x}_{s,T}$, with unknown parameter vector $\boldsymbol{\beta} = (1, \beta_1, \beta_2, \ldots, \beta_{T-1})'$. The trend information is desired as inputs of the SVC. A key challenge that we address is how to jointly estimate the parameter vectors $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$. We propose a novel longitudinal support vector classifier (LSVC) that jointly estimates the separating hyperplane parameters and the temporal trend parameters using quadratic programming. We present our approach using a simple linear kernel, but the ideas naturally extend to other kernel functions.

Let $\mathbf{X}_m^{\sim} = [\mathbf{X}_{t=1}^{\sim}, \mathbf{X}_{t=2}^{\sim}, \ldots, \mathbf{X}_{t=T}^{\sim}]'$ be a p by $T N$ matrix, with components $\mathbf{X}_{t=k}^{\sim} = (y_1 \boldsymbol{x}_{1,t=k}, y_2 \boldsymbol{x}_{2,t=k}, \ldots, y_N \mathbf{x}_{N,t=k})$ representing data from N subjects each with p features. The corresponding $\boldsymbol{\beta}_m$ is a $T N$ by N matrix.

$$\begin{aligned} \mathbf{G} &= \left(\tilde{\mathbf{X}}_{t=1} + \beta_1 \tilde{\mathbf{X}}_{t=2} + \ldots + \beta_{T-1} \tilde{\mathbf{X}}_{t=T}\right)^T \left(\tilde{\mathbf{X}}_{t=1} + \beta_1 \tilde{\mathbf{X}}_{t=2} + \ldots + \beta_{T-1} \tilde{\mathbf{X}}_{t=T}\right) \\ &= \left(\tilde{\mathbf{X}}_m \beta_m\right)^T \left(\tilde{\mathbf{X}}_m \beta_m\right) \\ &= \beta_m^T \mathbf{G}_m \beta_m, \end{aligned} \quad (2.4)$$

with

$$\mathbf{G}_m = \begin{bmatrix} \tilde{\mathbf{X}}_{t=1}^T \tilde{\mathbf{X}}_{t=1} & \cdots & \tilde{\mathbf{X}}_{t=1}^T \tilde{\mathbf{X}}_{t=T} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{X}}_{t=1}^T \tilde{\mathbf{X}}_{t=1} & \cdots & \tilde{\mathbf{X}}_{t=1}^T \tilde{\mathbf{X}}_{t=T} \end{bmatrix}$$

and $\beta_m^T = [\mathbf{I}_{N \times N}, \beta_1 \mathbf{I}_{N \times N}, \beta_2 \mathbf{I}_{N \times N}, \ldots, \beta_{T-1} \mathbf{I}_{N \times N}]$

Then, we denote $\boldsymbol{w}_{nv}$ as the estimate of separating hyperplane parameter in the classical SVC with inputs in the form of $\boldsymbol{x}_s = \boldsymbol{x}_{s,1} + \beta_1\boldsymbol{x}_{s,2} + \beta_2\boldsymbol{x}_{s,3}\ldots + \beta_{T-1}\boldsymbol{x}_{s,T}$. The primal objective function becomes

$$\min_{\boldsymbol{w}_{nv}} \frac{1}{2}\|\boldsymbol{w}_{nv}\|^2 + C\sum_{s=1}^{N}\xi_s \quad s=1,2,\ldots N \quad (2.5)$$

Similarly, with substituting $\boldsymbol{w}_{nv}=\sum_{s=1}^{N} y_s\alpha_s(\tilde{x}_s\beta_m)^T$, we can reparameterize the Langrange (Wolfe) dual function as:

$$\min_{\alpha} \frac{1}{2}\alpha_m^T\mathbf{G}_m\alpha_m - 1'\alpha \quad (2.6)$$

with subject to

$$C \geq \alpha_m(s) \geq 0,$$
$$\sum_{t}^{T}\sum_{s}^{N}\alpha_m(s+(t-1)N)\,y_s=0,$$
$$for \quad s=1,2,\ldots N \quad and \quad t=1,2,\ldots T-1.$$

In this way, the model augments the dimension of $\mathbf{G}_m$ to TN by TN and the augmented kernel is ensured to be semi-positive definite. After $\boldsymbol{\alpha}_m$ is determined, the separating hyperplane parameter becomes

$$\boldsymbol{w}_{nv}=\left[\sum_{s=1}^{n}\alpha_m(s)\,\boldsymbol{x}_{s,1}y_s+\sum_{s=1}^{n}\alpha_m(s+N)\,\boldsymbol{x}_{s,2}y_s,\ldots+\sum_{s=1}^{n}\alpha_m(s+N(T-1))\,\boldsymbol{x}_{s,T}y_s\right]. \quad (2.7)$$

Defining the $1 \times T$ vector $\boldsymbol{\alpha}_{m,s} = (\boldsymbol{\alpha}_m(s), \boldsymbol{\alpha}_m(s+N), \ldots, \boldsymbol{\alpha}_m(s+(T-1)N))$ we then have $\boldsymbol{w}_{n,v}=\Sigma_{s=1}^{n} y_s\alpha_{m,s}\tilde{x}_s$. In either case, we can notice that

$$\boldsymbol{w}_{nv}=\sum_{s=1}^{n} y_s\alpha_m(s)\left(\boldsymbol{x}_{s,1}+\beta_1\boldsymbol{x}_{s,2}+\beta_2\boldsymbol{x}_{s,3}\ldots+\beta_{T-1}\boldsymbol{x}_{s,T}\right).$$

After obtaining $\boldsymbol{w}_{nv}$, we have $b=\frac{1}{N}\sum_{s=1}^{N}\left(\boldsymbol{w}_{nv}\cdot(\tilde{x}_s\beta_m)^T - y_s\right)$, in which $\boldsymbol{\beta}_m$ can be estimated based on $\boldsymbol{\alpha}_m$. Hence, the separating hyperplane is

$$h(\tilde{\boldsymbol{x}}) = \boldsymbol{w}_{n,v}\cdot(\tilde{\boldsymbol{x}}\beta_m)^T + b. \quad (2.8)$$

The subjects with all $\boldsymbol{\alpha}_m > 0$ are considered as support vectors. Therefore, this method is different from directly applying SVC after stacking up the features at different times as

independent features. In fact, this naive expansion of the feature space is a special case of LSVC with all $\beta = 1$.

Besides estimating α and β vectors from α_m, we can alternatively employ an iterative procedure to estimate α and β with respect to an objective function of 2.6. The algorithm will take T quadratic programming steps for each iteration. We rewrite the first part of the objective function in 2.6 as:

$$\alpha \mathbf{G}_m^{0,0} \alpha + \alpha \beta_m^T \mathbf{G}_m^{0,T} \alpha + \alpha \mathbf{G}_m^{0,T} \beta_m \alpha + \alpha \mathbf{G}_m^{T,T} \beta_m \alpha, \quad (2.9)$$

where we denote

$$\mathbf{G}_m = \begin{bmatrix} \mathbf{G}_m^{0,0} & \mathbf{G}_m^{0,T} \\ \mathbf{G}_m^{T,0} & \mathbf{G}_m^{T,T} \end{bmatrix} \quad (2.10)$$

For example, $\mathbf{G}_m^{0,0}\,(N \times N)$ is the submatrix in the left top corner of the matrix $\mathbf{G}_m$ for the baseline data $\left(\tilde{\mathbf{X}}_{t=1}^T \tilde{\mathbf{X}}_{t=1}\right)$.

Since the sum of convex functions is still convex, we only need to prove that the objective function in 2.9 is convex with respect to α and β. We relegate the proof of convexity to the appendix. The convexity guarantees that the local minimum is also the global minimum and the solution for that minimum is unique. The algorithm is described as follows: (1) we start with initial values of β and use QP to optimize 2.9 to obtain α; (2) use the updated α obtained in step 1 and apply QP again to estimate β; (3) repeat the above two steps until convergence. The uniqueness of the solution leads to the convergence of the iterative algorithm.

2.3 Nonlinear Kernel Functions

Although the above derivations are considered in context of a linear kernel, it is natural to extend to nonlinear kernels. First, we can denote

$$\tilde{\mathbf{K}}\left(\tilde{x}_s, \tilde{x}_{s'}\right) = \begin{bmatrix} \mathbf{K}\left(\tilde{x}_{s,1}, \tilde{x}_{s',1}\right) & \cdots & \mathbf{K}\left(\tilde{x}_{s,1}, \tilde{x}_{s',T}\right) \\ \vdots & \ddots & \vdots \\ \mathbf{K}\left(\tilde{x}_{s,T}, \tilde{x}_{s',1}\right) & \cdots & \mathbf{K}\left(\tilde{x}_{s,T}, \tilde{x}_{s',T}\right) \end{bmatrix} \quad (2.11)$$

and we have $< \beta \mathbf{K}(\cdot, \tilde{x_{s,t}}), \mathbf{K}(\cdot, \tilde{x_{s',t}}) >= \beta \mathbf{K}(\tilde{x_{s,t}}, , \tilde{x_{s',t}})$, where $\mathbf{K}(\cdot, \tilde{x_{s,t}})$ indicates the reproducing kernel map of $\tilde{x_{s,t}}$. Therefore, the temporal trend is taken on the reproducing kernel mapped space which may be a set of nonlinear transformations of $\tilde{x_{s,t}}$, say $\mathbf{K}(\cdot, \tilde{x_{s,t=0}}) + \beta \mathbf{K}(\cdot, \tilde{x_{s,t=1}}) .. , + \beta_{T-1} \mathbf{K}(\cdot, \tilde{x_{s,t=T-1}})$.

Thus, the reproducing kernel function of separating hyperplane becomes

$$h\left(\tilde{\boldsymbol{x}}\right)=\sum_{s=1}^{N} y_s \alpha_m \tilde{\mathbf{K}}\left(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_s\right) \beta_m^T + b, \quad (2.12)$$

where b is obtained by $b=\frac{1}{N}\sum_{s=1}^{N}\sum_{s'=1}^{N}\left(y_s y_{s'} \alpha_m \tilde{\mathbf{K}}\left(\tilde{x}_s, \tilde{x}_{s'}\right) \beta_m^T - y_s\right)$. In this way, the temporal trend parameter vector's length is increased in accordance with the dimension of features mapped. Hence, it also could be considered to estimate nonlinear temporal trends of the original features.

2.4 Feature Selection and Parameter Tuning

Feature selection is a critical step in supervised learning, as it can reduce the dimensionality of the feature space, leading to increased robustness, improved stability of the classifier, and reduced computational load. However, for longitudinal HDD feature selection based on 'filtering' may not be applicable because elements of $\boldsymbol{\beta}$ are unknown and thus no statistical test can be conducted for each feature. Nevertheless, 'wrapper' procedures such as SVC based recursive feature elimination (RFE) algorithm is valid under our new LSVC model. The SVC-RFE algorithm was first proposed by Guyon *et al*., 2003, and it ranks all the features according to a classifier based weight function and eliminates one or more features with the lowest weights. This process is repeated until the minimal set of features achieve high classification accuracy. For a linear SVC, the weights are simply summarized from the $p \times 1$ vector $\boldsymbol{w}$. For non-linear kernel SVC, the rank of a feature is determined by the impact that its removal has on the variation of $\mathrm{I}\boldsymbol{w}\mathrm{I}^2$. In context of longitudinal HDD, the rank is determined by:

$$\left|\|\boldsymbol{w}_{nv}\|^2 - \|\boldsymbol{w}_{nv}^{-v}\|^2\right| = \left|\alpha_m^T \mathbf{G}_m \alpha_m - \left(\alpha_m^{-v}\right)^T \mathbf{G}_m^{-v} \alpha_m^{-v}\right|, \quad (2.13)$$

where $\boldsymbol{w}_{nv}^{-v}, \alpha_m^{-v}, \quad and \quad \mathbf{G}_m^{-v}$ are the estimates and inputs without feature or features v.

In addition, the tuning parameters such as cost C are also important, as they can affect the estimate of separating hyperplane parameters as well as temporal trend parameters. If we consider C as the level of shrinkage, and large C corresponds to light regularization and small C stands for heavy regularization. Therefore, we can use the SVC path algorithm by starting with large C (low regularization) and increase it gradually, and observe the path of shrinkage in terms of $\boldsymbol{\alpha}_m(C)$ (Hastie and Tibshirani, 2004). This process will provide insight concerning the bias and variance trade off. For LSVC, we can utilize this shrinkage path algorithm to better estimate the temporal trend parameters.

3 Results

We investigate the performance of the proposed method by using simulation data and by using data from a longitudinal neuroimaging study.

3.1 Simulation study

To evaluate the performance of our proposed LSVC, we generate longitudinal data for 200 subjects and evenly divide them into two groups. Data for each subject includes $p = 100$ features at two time points ($T = 2$). We generate a group label $y_s \in \{-1, 1\}$ and features $\mathbf{x}_s$ for each subject. We also use a binary variable z to determine the baseline feature expression level, e.g. if $z_s = 1$ then $\mathbf{x}_s = \mathbf{1}$ otherwise $\mathbf{x}_s = \mathbf{0}$. Within each group, half of of the subjects have $z_s = 1$ at the lowest level of separability of the baseline data. If $z_s = y_s$, the baseline data are 100% separable. We then set up the temporal change variable Δ that depends on the group label y by letting $\Delta_s = \mathbf{1}$, if $y_s = 1$, otherwise $\Delta_s = \mathbf{0}$. Thus, different groups have different temporal trends. Therefore, the simulation is generated as follows:

$$\begin{aligned}
\mathbf{x}_{s,t=1}|z=0 &N\left(\mathbf{0}, \mathbf{I}\cdot\sigma^{\mathbf{2}}\right), \\
\mathbf{x}_{s,t=1}|z=1 &N\left(\mathbf{1}, \mathbf{I}\cdot\sigma^{\mathbf{2}}\right), \\
\Delta_s|y=-1 &N\left(\mathbf{0}, \mathbf{I}\cdot\tau^{\mathbf{2}}\right), \\
\Delta_s|y=1 &N\left(\mathbf{1}, \mathbf{I}\cdot\tau^{\mathbf{2}}\right),
\end{aligned}$$

and $\mathbf{x}_{s,t=2} = \mathbf{x}_{s,t=1} + a \cdot \Delta$, where a is scalar to denote the magnitude and direction of the change. In this simulation, we use $\sigma^2 = 0.01$, $\tau^2 = 0.001$, and $a = 1$ to generate the data. The generated data is depicted in Figure 1, with the x -axis indicating the subject number (the first 100 subjects are in group one, the rest are in group two), and the y -axis indicating the feature expression level. The three subplots describe baseline, time one, and the temporal trend.

We test the performance of the model using different parameter and separability conditions. The variance has little influence on the model if the data are not separable, but separability definitions do impact the results. Therefore, we consider four methods: SVC based on baseline data, SVC based on both baseline and time one data stacked and treated as independent (i.e. no temporal trends), our proposed LSVC, and SVC with a known trend. We test these methods using separability levels of 50 %, 60 % and 70%. The separability level between groups could also be considered as a function of the variation between subjects within each group, where a lower level of separability between groups results from higher between-subjects (in one group) variation. Also, we run an additional simulation by introducing different random ”subject” effects upon the features both at baseline and time 1. The random effects are assumed to have zero mean and variance σ^2, $2\sigma^2$, and $5\sigma^2$ (totally blurring), and the higher level of noise leads to lower level of SNR ratio. We then evaluate the performances of the classifiers under different levels of SNR ratios (see table 2). For all cases, we only consider linear kernels for equitable comparisons. In addition, for the tuning parameter C there is no closed form estimator though cross validation can be applied to assist in determining the best-performing value of C from a pre-specified list of values (Hastie and Tibshirani, 2004). We feel that generating prediction results based on different levels of C provides a better evaluation of the SVC’s performance when comparing different models.

We present the accuracy results (and standard deviation) for each method and for each simulation setting in Table 1. The results indicate that our LSVC has excellent performance, which is comparable to the 'oracle' model with perfect accuracy in our simulation example. Here, SVC with 'oracle' represents the SVC as if the temporal trend is known and maximal information is obtained for LHDD. The traditional SVC performs very poorly across all simulation settings.

Similarly, Table 2 shows the results for the simulated data with 50% separability and different levels of noise. When the noise level is low, LSVC performs better than traditional methods and approximates the 'oracle' model. When we double the original noise level considered, i.e. the noise is taken to be $2\sigma^2$, the LSVC still performs quite well and shows marked improvements over the traditional SVC. We also consider a case where the noise level saturates the signal, specifically $5\sigma^2$. Although this case is not likely to arise in practice, we wanted to evaluate the performance of our method under extreme conditions. Naturally, the performance of our model declines in this setting, but it still outperforms the conventional SVC approaches.

3.2 Data Example

We analyze data from the ADNI database (www.loni.ucla.edu/ADNI), which includes longitudinal PET scans acquired at baseline, 6 months, and 12 months. We used data from 80 subjects, 40 Alzheimer's disease (AD) patients and 40 healthy controls, ages 62 to 84. We used SPM5 for data preprocessing. We illustrate our longitudinal SVC procedure using PET scans from baseline and 12 months.

We use 1877 voxels within AD relevant regions of interest (ROI) as features, for example the hippocampus and entorhinal cortex (see Figure 2). Based on voxels within selected ROIs, we applied our novel longitudinal SVC to discriminate healthy and AD groups. Our goal here is not to chase perfect classification of accuracy through tuning parameters and feature selection, rather we demonstrate the usage of the proposed method and compare with the alternatives. For the validation procedure, we choose leave one out cross-validations. We tested on the data by using three classification methods SVC: with baseline session only, SVC with two sessions stacked independently (N by $2p$); and the proposed LSVC. Also, different kernels are used. The results show that the accuracies for the two alternative methods across all costs are around 50% when polynomial (degree 2, 3, 5, 10) and Gaussian kernels (with various values of σ) are used. We tune the cost parameter across C = (0.1, 1, 100, 10000). The accuracies are listed in Table 3 based on a leave-one-out cross validation across all costs. In general the accuracy of LSVC method is 10 to 15 percent higher than the other two alternative methods.

Overall, based on the simulation study and neuroimaging data analysis, our proposed method outperforms the traditional methods.

4 Discussion

In this article, we present a novel support vector classifier for LHDD. Our proposed method estimates decision function parameters and longitudinal parameters simultaneously using

quadratic programming. The classifier can be extended to any kernel that satisfies Mercer's condition, and then the temporal trend is based on the nonlinear transformations of the original feature space. The SVC-RFE feature selection procedure can also be conducted in our LSVC, with ranking weight based on the width of the separating margins.

We apply the proposed method to longitudinal neuroimaging data which is a type of LHDD with temporal and spatial correlation structure. A growing literature has addressed the issue of temporal and spatial correlation when modeling neuroimaging data as dependent variables (Bowman *et al*., 2008, Derado *et al*., 2010). However, in our model the LHDD represent independent variables, and the group label for each subject is the dependent variable, and usually we do not explicitly account for correlations of the predictors. Note that we model the temporal trend for the LHDD to account for the temporal correlations introduced by the longitudinal experimental design. For fMRI data, since we use the 1st level analysis results as features, the scan to scan temporal correlation is considered in the first level analysis using conventional approaches such as prewhitening or precoloring.

In our data example, we use the biological information to effectively reduce the number of features from around 300,000 to 1877, rather than performing variable selection empirically. When such biological information is not present, some supervised methods are applicable. Based on the results from our simulation study and our data example, the LSVC leverages the additional information from longitudinal measurements to achieve higher prediction accuracy. The computational load of our LSVC technique is generally quite manageable, and on average training a LSVC model of 200 subjects with 100 features takes roughly 14 minutes on a PC with Intel Core2 Duo 2.83G CPU and 4G memory.

Acknowledgments

This research was supported by NIH grants R01-MH079251 (PI. Dr. F DuBois Bowman).

5 Appendix: Proof

Proposition. $f=\alpha_m^T \mathbf{G}_m \alpha_m$ is a convex function regarding $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, where $\boldsymbol{\alpha}_m = (\boldsymbol{\alpha}, \beta_1\boldsymbol{\alpha}, \ldots, \beta_{T-1}\boldsymbol{\alpha})$.

Proof. The second order condition of convexity requires the Hessian matrix $\nabla^2 f$ to be positive semidefinite (p.s.d.). and

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial \alpha^2}, & \frac{\partial^2 f}{\partial \alpha \partial \beta} \\ \frac{\partial^2 f}{\partial \beta \partial \alpha}, & \frac{\partial^2 f}{\partial \beta^2} \end{pmatrix}$$

. Here first present the case of two time points and extend it to *T* time points. Therefore,

$$f=\begin{pmatrix} \alpha \\ \beta\alpha \end{pmatrix}^T \begin{bmatrix} \mathbf{G}_m^{0,0} & \mathbf{G}_m^{0,1} \\ \mathbf{G}_m^{1,0} & \mathbf{G}_m^{1,1} \end{bmatrix} \begin{pmatrix} \alpha \\ \beta\alpha \end{pmatrix}$$

where $\mathbf{G}_m^{0,0}=\tilde{\mathbf{X}}_{t=1}^T\tilde{\mathbf{X}}_{t=1}$, $\mathbf{G}_m^{0,1}=\tilde{\mathbf{X}}_{t=1}^T\tilde{\mathbf{X}}_{t=2}$, $\mathbf{G}_m^{1,0}=\tilde{\mathbf{X}}_{t=2}^T\tilde{\mathbf{X}}_{t=1}$ *and* $\mathbf{G}_m^{1,1}=\tilde{\mathbf{X}}_{t=2}^T\tilde{\mathbf{X}}_{t=2}$.

Then, the four derivatives are:

$$\begin{aligned} \tfrac{\partial^2 f}{\partial\alpha^2}_{(N\times N)} &= \mathbf{G}_m^{0,0}+\beta\mathbf{G}_m^{0,1}+\beta\mathbf{G}_m^{1,0}+\beta^2\mathbf{G}_m^{1,1} \\ \tfrac{\partial^2 f}{\partial\alpha\partial\beta}_{(N\times 1)} &= \left(\mathbf{G}_m^{0,1}+\beta\mathbf{G}_m^{1,1}\right)\alpha \\ \tfrac{\partial^2 f}{\partial\beta\partial\alpha}_{(1\times N)} &= \alpha^T\left(\mathbf{G}_m^{0,1}+\beta\mathbf{G}_m^{1,1}\right) \\ \tfrac{\partial^2 f}{\partial\beta^2}_{(1\times 1)} &= \alpha^T\mathbf{G}_m^{1,1}\alpha \end{aligned}$$

Next, we need to prove $\nabla^2 f$ is p.s.d.. For any nonzero vector $\mathbf{v}$ of length N and scalar u,

$$\begin{pmatrix} \mathbf{v} \\ u \end{pmatrix}^T \begin{pmatrix} \frac{\partial^2 f}{\partial\alpha^2}, & \frac{\partial^2 f}{\partial\alpha\partial\beta} \\ \frac{\partial^2 f}{\partial\beta\partial\alpha}, & \frac{\partial^2 f}{\partial\beta^2} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ u \end{pmatrix}$$

$$\begin{aligned} &=\mathbf{v}^T\mathbf{G}_m^{0,0}\mathbf{v}+\mathbf{v}^T\mathbf{G}_m^{0,1}\alpha u+\beta\mathbf{v}^T\mathbf{G}_m^{0,1}\mathbf{v}+u\alpha^T\mathbf{G}_m^{0,1}\mathbf{v}+\mathbf{v}^T\mathbf{G}_m^{1,0}\mathbf{v}\beta \\ &+u\alpha^T\mathbf{G}_m^{1,1}\mathbf{v}\beta+\beta\mathbf{v}^T\mathbf{G}_m^{1,1}\alpha u+\beta\mathbf{v}^T\mathbf{G}_m^{1,1}\mathbf{v}\beta+u\alpha^T\mathbf{G}_m^{1,1}\alpha u \\ &=\left[\tilde{\mathbf{X}}_{t=2}\left(\beta\mathbf{v}+u\alpha\right)+\tilde{\mathbf{X}}_{t=1}\mathbf{v}\right]^T\left[\tilde{\mathbf{X}}_{t=2}\left(\beta\mathbf{v}+u\alpha\right)+\tilde{\mathbf{X}}_{t=1}\mathbf{v}\right]+\beta\mathbf{v}^T\mathbf{G}_m^{1,1}\mathbf{v}\beta+u\alpha^T\mathbf{G}_m^{1,1}\alpha u\geq 0 \end{aligned}$$

because $\mathbf{G}_m^{1,1}$ is p.s.d..

Similarly for T time points data set, the Hessian matrix

$$\nabla^2 f=\left[\tilde{\mathbf{X}}_{t=1}\mathbf{v}+\sum_{k=1}^{T-1}\tilde{\mathbf{X}}_{t=k+1}\left(\beta_k\mathbf{v}+u_k\alpha\right)\right]^T\left[\tilde{\mathbf{X}}_{t=1}\mathbf{v}+\sum_{k=1}^{T-1}\tilde{\mathbf{X}}_{t=k+1}\left(\beta_k\mathbf{v}+u_k\alpha\right)+\sum_{k=1}^{T-1}u_k\alpha^T\mathbf{G}_m^{k,k}\alpha u_k+\sum_{k=1}^{T-1}\beta_k\mathbf{v}^T\mathbf{G}_m^{1,1}\mathbf{v}\beta_k\right]$$

is also p.s.d..

Moreover, the objective functions with nonlinear kernels are also convex if each $\mathbf{K}(\tilde{\mathbf{X}_{t=k}}, \tilde{\mathbf{X}_{t=k}})$ follows Mercer's condition and is p.s.d. for $k = 1, 2, \ldots, T$.

References

Mitchell TM, Hutchinson R, Niculescu RS, Pereira F, Wang X, Just M, Newman S. Learning to decode cognitive states from brain images. Machine Learning. 2004; 57:145–175.

LaConte S, Strother S, Cherkassky V, Anderson J, Hu X. Support vector machines for temporal classification of block design fMRI data. Neuroimage. 2005; 26:317–329. [PubMed: 15907293]

Chen S, Hong D, Shyr Y. Wavelets-based Procedures for Proteomic Mass Spectrometry Data Processing. Computational Statistics and Data Analysis. 2007; 52:211–220.

Mourao-Miranda J, Bokde AL, Born C, Hampel H, Stetter M. Classifying brain states and determining the discriminating activation patterns: support vector machine on functional MRI data. Neuroimage. 2005; 28:980–995. [PubMed: 16275139]

Fu CHY, Mourao-Miranda J, Costafreda SG, Khanna A, Marquand AF, Williams SCR, Brammer MJ. Pattern classification of sad facial processing: toward the development of neurobiological markers in depression. Biol Psychiatry. 2008; 63:656–662. [PubMed: 17949689]

Vapnik, V. Statistical Learning Theory. Wiley; 1998.

Bowman F, Caffo B, Bassett S, Kilts C. A Bayesian hierarchical framework for spatial modeling of fMRI data. NeuroImage. 2008; 39:146–156. [PubMed: 17936016]

Derado G, Bowman F, Kilts C. Modeling the spatial and temporal dependence in fMRI data. Biometrics. 2010; 66:949–957. [PubMed: 19912175]

Craddock RC, Holtzheimer PE, Hu XP, Mayberg HC. Disease State Prediction From Resting State Functional Connectivity. M agnetic Resonance in Medicine. 2009; 62:1619–28.

Vapnik, V. The nature of statistical learning theory. Springer; New York: 1996. p. 188

Guyon I, Elisseeff A. Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3. 2003:1157–1182.

Wahba, G. Spline Models for Observational Data. SIAM; Philadelphia, PA: 1990.

Hastie, T.; Tibshirani, R. Generalized Additive Models. Chapman and Hall; 1990.

Hastie T, Rosset S, Tibshirani R, Zhu J. The Entire Regularization Path for the Support Vector Machine. Journal of Machine Learning Research. 52004:1391–1415.

Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of American Statistical Association. 2001:1348–1360.

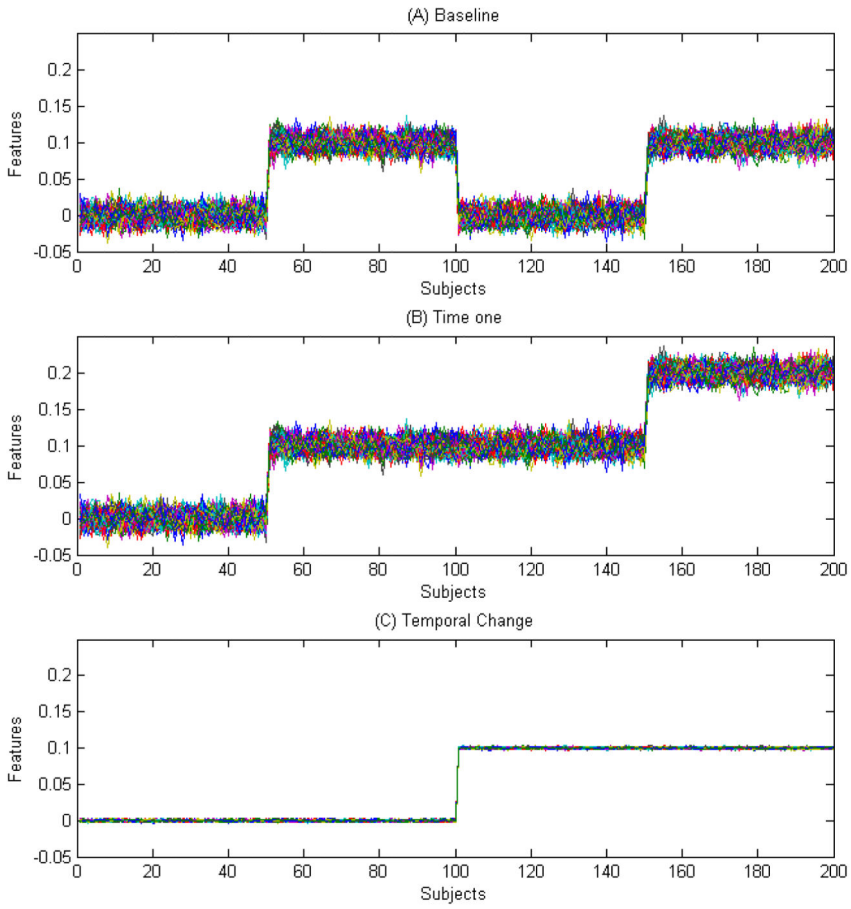

Figure 1.
Simulated Data Set: (A) Baseline data, (B) Time one data, and (C) Temporal change

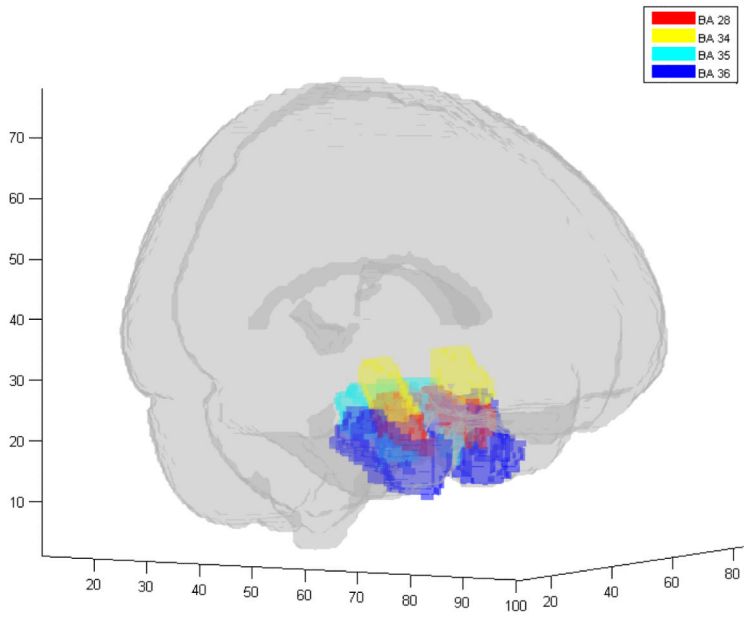

Figure 2.
Voxels in these ROIs are used for analysis

Table 1

Simulation Classification Results with Different Separability

Cost (C)	SVC baseline	SVC stack	LSVC	SVC "oracle"
50% separable at baseline				
0.1	.49 (0.06)	.51 (0.04)	.99 (0.01)	1(0.0)
1	.52 (0.03)	.50 (0.03)	1 (0.00)	1(0.0)
100	.50 (0.02)	.53 (0.04)	1 (0.01)	1(0.0)
10000	.53 (0.03)	.48 (0.06)	.99 (0.03)	1(0.0)
60% separable at baseline				
0.1	.57 (0.16)	.71 (0.31)	1 (0.0)	1(0.0)
1	.52 (0.23)	.75 (0.11)	1 (0.0)	1(0.0)
100	.58 (0.12)	.73 (0.14)	1 (0.01)	1(0.0)
10000	.63 (0.08)	.72 (0.26)	1 (0.02)	1(0.0)
70% separable at baseline				
0.1	.72 (0.13)	.83 (0.04)	1 (0.01)	1(0.0)
1	.78 (0.07)	.87 (0.03)	1 (0.02)	1(0.0)
100	.73 (0.12)	.82 (0.04)	1 (0.02)	1(0.0)
10000	.71 (0.21)	.81 (0.06)	1 (0.06)	1(0.0)

Table 2

Simulation Classification Results with Different Noise Level

Cost (C)	SVC baseline	SVC stack	LSVC	SVC "oracle"
		Noise: σ^2		
0.1	.47 (0.13)	.54 (0.07)	.98 (0.03)	1(0.0)
1	.56 (0.09)	.50 (0.12)	.99 (0.01)	1(0.0)
100	.51 (0.06)	.61 (0.14)	.99 (0.01)	1(0.0)
10000	.52 (0.10)	.55 (0.08)	.99 (0.01)	1(0.0)
		Noise: $\sigma 2^2$		
0.1	.57 (0.16)	.55 (0.21)	.96 (0.03)	1(0.0)
1	.52 (0.23)	.55 (0.14)	.96 (0.02)	1(0.0)
100	.48 (0.12)	.52 (0.18)	.98 (0.01)	1(0.0)
10000	.55 (0.08)	.49 (0.21)	.97 (0.02)	1(0.0)
		Noise: $5\sigma^2$		
0.1	.48 (0.33)	.47 (0.28)	.56 (0.22)	.73 (0.11)
1	.54 (0.17)	.52 (0.23)	.61 (0.18)	.68 (0.20)
100	.53 (0.22)	.51 (0.26)	.54 (0.15)	.70 (0.17)
10000	.51 (0.24)	.49 (0.16)	.58 (0.06)	.62 (0.13)

Table 3

ADNI PET Data Classification Results

Cost (C)	SVC baseline	SVC stack	LSVC
0.1	.65	.66	.78
1	.66	.67	.76
100	.65	.67	.75
10000	.66	.66	.75