Support Vector Machines for Longitudinal Analysis

Kristiaan Pelckmans, Hong-Li Zeng

Abstract—This paper considers the problem of learning a classifier from observed longitudinal data. Here, each data-point takes the form of a single time-series. Assuming that each such series comes with a binary label, the problem of learning to score this label of a new time-series is considered. Here, the notion of margin underlying the classical Support Vector Machine (SVM) is extended to LSVM for functional data. LSVM is also a convex optimization problem and its dual form is derived. Empirical results for specified cases with significance tests indicate the efficacy of the LSVM for analyzing such longitudinal data.1

Index Terms—Classification, Longitudinal Data Analysis, Support Vector Machines, Large Margin Analysis, Multiple Testing.

I. INTRODUCTION

LONGITUDINAL or functional data consists of time-series rather than of single samples. Typical examples are found in growth curves, in signals emanating from different celestial objects, or in the analyses of medical trials where patients are followed for a period of time, see e.g. [1], [2]. The traditional way to analyze this data is by inferring a stochastic model explaining the data, making use of generalized linear models or random effect models. However, such approaches are typically constrained by a curse of dimensionality [3], [4], yielding difficulties when presented with high-dimensional data as for example resulting from image processing.

This paper explores a technique which is capable of dealing with such data, by following the lines of thinking as set out in the hugely successful literature on Support Vector Machines (SVMs) [5], [6], [7]. The power of SVMs is generally attributed to three different foundations: (i) the use of the computationally practical and theoretical well-understood device of convex optimization, (ii) construction of nonlinear models by using representer kernels and high-dimensional linear approaches, (iii) a solid foundation in learning theory. We will focus in this paper on (i) and (iii), since nonlinear extensions - although straightforward - are not relevant for the computational and theoretical well-understood

The goal of this analyses differs somewhat from what is aimed for by statistical analysis of longitudinal analysis. As pointed out in the introduction of [2], traditionally one aims for

- The impact of the deterministic covariates on the responses.
- The presence and relevance of random effects underlying the observations.
- The form of the autocovariance function underlying the responses.

Here however, the aim is slightly different: the objective is to find those features which evolve mostly in different ways in-between the two classes. This is known as longitudinal data classification [8], or functional discriminant analysis [9], [10]. Marshall et al. in [8] describe a linear/quadratic discriminant analysis (LDA/QDA) for functional data. The means of the classes, as well as they covariance matrices that determine the class labels are estimated by an iteration process. James and Hastie proposed a functional linear discriminant analysis (FLDA) [9] algorithm by extending the LDA method aiming at dealing with the irregular curves. Here, there are no explicit covariates, but the longitudinal data itself plays this role. The response variable are in general multivariate, and one seeks to find the feature of those responses which differs at most in-between the two classes. This aim goes much along the lines as taken by SVMs for classification, and it is exactly this correspondence that this paper explores in some detail.

A handful of authors have considered the analysis of longitudinal data with SVM-like methods before. For example, Suykens et al. presented a least square support vector machine [11], which is a modified version of SVM that contains the least squares cost function and equality instead of inequality constraints. Luts et al. extended the LS-SVM to a mixed effects LS-SVM model [12] aiming at classifying longitudinal data. Chen and Brown [13] developed a Longitudinal Support Vector Classifier (LSVC) which extends the binary SVM to deal with functional high dimensional dataset, basically by stacking the time-series in one large input vector. Such approach is essentially bound to regularly sampled time-series. Biau et al. [14] give a theoretical account of classification with longitudinal data using a nearest neighbour approach. Methods of machine learning have been widely applied to image processing, notably for medical imaging. Wernick et al. surveys the field in [15].

The approach presented here is tailored to the case of longitudinal analysis where the different subjects originate from a common pool. That is, in the beginning of the experiment, there is little hope that the different classes can be separated. Further up in the experiment however, it is reasonable to expect that the different classes present a growing difference. Secondly, this approach is tailored towards the case where the different subjects are sampled at different time instances ('irregular sampling'). Note that in the case where all subjects are sampled at shared times ('regular sampling'), one can reduce this application to a standard classification task, as done in [13]. In order to cope with irregular sampling, the idea of a (parametric) margin function is introduced. That is,
samples and their labels are contrasted to the margin function evaluated at the relevant time instances. This work focuses on the use of linear margin functions since they give a good trade-off between mathematical convenience and usefulness in the application mind. Besides irregular sampling, margin function and tailoring to the evolutionary setup, this paper derives a convex program which is used to compute efficiently the solution.

The application motivating the development of this method goes as follows. The aim is to classify breeding lines of an artificial selection experiment. The brains of the individuals in those breeding lines are scanned, and the LSVM is used to detect differences in those for different selection regimes. In a prototypical case, two classes of breeding lines are considered. The first class evolves according to ‘no selection’ and serves as a control. The second class imposes a selection bias in the evolutionary setup. That is, in the beginning of the processes the two classes are overlapping. They gradually diverge only when time (evolution) goes on.

This paper is organized as follows. The next section details the proposed formulation. Section III provides details on the evolutionary setup. That is, in the beginning of the processes the two classes are overlapping. They gradually diverge only when time (evolution) goes on. Section IV exemplifies the approach on artificially created case studies. Section V concludes this paper.

II. THE LONGITUDINAL SUPPORT VECTOR MACHINE (LSVM)

This section gives the proposed learning scheme. The following setup is adopted. Let $X_i(t)$ denote a characterization of the $i$th site at time $t$. In the remainder of this paper, $X_i(t)$ is a random variable which takes values in $\mathbb{R}^p$ with $p > 0$. The $i = 1, \ldots , n$ subjects are divided into two classes, hereto we reindex the subjects such that $X_i^\up (t)$ belong to the first class for $i = 1, \ldots , n_+$, and $X_i^- (t)$ to the second one for $i = 1, \ldots , n_-$ so that $n = n_+ + n_-$. Equivalently, let $y_i = \{-1, 1\}$ the label corresponding to $X_i$. Then the goal is to find vectors $w_+, w_- \in \mathbb{R}^p$ such that

\[ \begin{align*}
  w_+^T X_i^+(t) + b & \geq 0 \quad \forall i = 1, \ldots , n_+ \\
  w_-^T X_i^-(t) + b & < 0 \quad \forall i = 1, \ldots , n_- 
\end{align*} \tag{1} \]

In case such a vectors exists, there are in general multiple solutions possible. The key idea is to single one solution out. Therefore, an additional principle is needed.

Let us start with the cross-sectional case, i.e. the case for fixed $t$. Fix $t \geq 0$ and consider the two sets $\{X_i^+(t)\}_i$ and $\{X_i^-(t)\}_i$. Then the maximal margin classifier or the hard SVM solves

\[ \begin{align*}
  (\hat{w}_t, \hat{m}_t) = \arg \max_{w_t, m_t} m_t \\
  \text{s.t.} \begin{cases} 
    w_t^T X_i^+(t) \geq m_t & \forall i = 1, \ldots , n_+ \\
    w_t^T X_i^-(t) \leq -m_t & \forall i = 1, \ldots , n_- \\
    \|w_t\|_2 = 1.
  \end{cases} \tag{2}
\]

Here, $m_t$ denotes the size of the margin at time $t$. This problem can be converted into a convex Quadratic Program (QP) using a change of variables.

Now, this can be done for any $t$, while it is natural to impose to have the same vector $w_t = w$. The question is of what to do with the margins $\{m_t\}_t$. Rather than having independent constants $m_t$ for every $t$, a parametric model of the margin function is imposed: $m_t = at + d$,

\[ m_t = at + d, \tag{3} \]

with parameters $(a, d)$. Note that this function can go negative, resulting in a negative margin: this means that samples from both classes can take the same values. This is a natural feat when the two classes have no clear distinction: this typically happens in the begin phase of the longitudinal study when the differentiating process has only started.

One very attractive feature of this formulation is that this approach can cope naturally with irregularly sampled data. That is, every time-instant that a sample is available, it is checked for with the margin function. Formally, let each subject $X_i$ for $i = 1, \ldots , n$ be sampled $n_i$ times $t_{ij} \leq \cdots \leq t_{i_{n_i}}$. The resulting classifier becomes

\[ \begin{align*}
  (\hat{w}, \hat{a}, \hat{d}) = \arg \max_{w,a,d} \sum_{i=1}^n \sum_{j=1}^{n_i} (at_{ij} + d) \\
  \text{s.t.} \begin{cases} 
    y_i (w^T X_i(t_{ij}) + b) \geq at_{ij} + d & \forall i = 1, \ldots , n, \forall j = 1, \ldots , n_i \\
    \|w\|_2 = 1,
  \end{cases} \tag{4}
\]

where the label of a new observation $X*$ which is measured at time points $t_1 \leq \cdots \leq t_{n_\ast}$, is predicted as

\[ \begin{align*}
  y_* = +1 & \text{ if } \sum_{j=1}^{n_\ast} (w^T X_\ast(t_{ij}) + b) \geq 0 \\
  y_* = -1 & \text{ if } \sum_{j=1}^{n_\ast} (w^T X_\ast(t_{ij}) + b) > 0. \tag{5}
\end{align*} \]

Note that this formulation aggregates the margins by using $\sum_{j=1}^{n_\ast}$. While this choice is ad-hoc at this point, this average margins idea comes with desirable properties as shown later.

Observe that $w$ will always be taken as large as possible, such that one can safely replace the constrain $\|w\|_2 = 1$ by $\|w\|_2 \leq 1$. However, there is a slight problem when the overall average margins is at best negative, then a $\|w\|_2 = 0$ would be favoured. This is to be checked case-by-case. Secondly, we introduce slack variable $\{\epsilon_{ij}\}_i$, making it possible to violate the margin in a small number of cases, increasing robustness of the algorithm. This results in the following convex QP:

\[ \begin{align*}
  \max_{w,a,d,\{\epsilon_{ij}\}} \sum_{i=1}^n \sum_{j=1}^{n_i} (at_{ij} + d) - C \sum_{i=1}^n \sum_{j=1}^{n_i} \epsilon_{ij} \\
  \text{s.t.} \begin{cases} 
    y_i (w^T X_i(t_{ij}) + b) \geq at_{ij} + d - \epsilon_{ij} & \forall i = 1, \ldots , n, \forall j = 1, \ldots , n_i \\
    \epsilon_{ij} \geq 0 & \forall i = 1, \ldots , n, \forall j = 1, \ldots , n_i \\
    \|w\|_2 \leq 1,
  \end{cases} \tag{6}
\]

where $C \geq 0$ is a user-defined (tuning) parameter. The formulation eq. (6) is convex, and can be solved efficiently using
polynomial time solvers for convex Quadratically Constraint Linear Programs (QCLPs). Appendix A derives the dual of this problem, which is computationally much more attractive to solve in case of large $p$.

It is useful to note how this formulation copes with the themes traditionally addressed in a statistical analysis of such data. First of all, the LSVM only aims to identify the difference between the two classes. That is, we do not aim to model the dynamics underlying the individual observation, and there is no need to recover the autocovariance functions. This would be too difficult a task anyways for applications where $p \gg n$, since the covariance function would be of order $p^2$.

Secondly, this formulation deals with random effects of the $n$ different subjects as follows. The inequality of the margin merely takes into account the worst case at any given time-point. (In case of $C < \infty$, one relaxes effectively to approximately worst case). That is, random effects are taken into account by using ‘$\leq$’ rather than ‘$=$’ in the constraints of (6). Again, there is no intention to recover the random effects, only to infer a global (i.e. population) property differentiating the two classes.

III. THE MARGIN FUNCTION

This section addresses the question why the ‘average margins’ is a good idea. At first, it translates the intuition as to how the two classes drift apart in the envisioned application. But it turns out that invoking ‘$\approx$’ is a good idea. At first, it translates the intuition as to how.

This is the learning-theoretical foundation of the use and design of SVMs for classification, fundamental in the work of Vapnik, see e.g. his IEEE overview paper [7] and the references therein. Recent advances in VC theory for large-margin classifiers are summarised and extended in [17].

The longitudinal problem is reduced to this case as follows. For ease of notation, let $n_1 = \ldots = n_m = n$ and fix constants $t_1 \leq \ldots \leq t_n$ at which times all $m$ subjects are measured, that is $t_1 = t_{11} = \ldots = t_{m1}, t_i = \ldots$ and $t_n = t_{1n} = \ldots = t_{mn}$. Hence the deterministic average-margin is given as

$$m(a, d) = \frac{2a}{n} \sum_{j=1}^n t_j + 2d.$$  
(12)

The decision functions are given as

$$f(X; w) = I \left( \sum_{j=1}^n w^T X(t_j) \geq 0 \right).$$  
(13)

So, one has that

$$\mathcal{F}_m = \{(f(X_1; w), \ldots, f(X_m; w)), w : \|w\|_2 \leq 1 \}. \quad (14)$$

The size of this set is bounded when only considering functions realizing a certain average-margin:

**Corollary 1:** Let $t_j \leq \ldots \leq t_n$ be fixed constants as defined before. Consider the functions $\mathcal{F}(\mu)$ realizing a minimal average-margin $\mu > 0$ on the sample. The corresponding set $\mathcal{F}_m(\mu)$ has a VC dimension bounded by

$$h \leq \min \left( m + 1, \left[ \frac{n^2 r^2}{\mu^2} \right] \right),$$  
(15)

where $\mu \geq m(a, d)$ is defined as in (12) and $r \geq \max_{i,j} \|X_i(t_j)\|_2$ almost surely.

This follows immediately by the following reduction. Let

$$X_i = (X_i(t_1)^T, \ldots, X_i(t_n)^T)^T \in \mathbb{R}^{pn}, \quad (16)$$

and

$$\tilde{w} = (w^T, \ldots, w^T)^T \in \mathbb{R}^{pn}. \quad (17)$$

Then $I(\tilde{x}^T \tilde{w} \geq 0)$ is equivalent to (13), and it is a classical, linear learning rule. Moreover, it has margin $\frac{\mu}{2} \sum_j d_{t_j} \geq \frac{\mu}{2} \sum_{t} a_{t_j} + 2d$, and radius bounded by $nr$. Observe that the $\frac{1}{2}$ factor comes in as $\|w\|_2 \leq n$ as $\|w\|_2 \leq 1$. Hence the radius-margin bound of Theorem 1 applies directly. Note also that if for each $t_j, h_j$ cases can be shattered with margin $b_j$ by \{w^T : \|w\|_2 \leq 1\}, one can also shatter with $\mathcal{F}_m$ at least $m \geq \min_j h_j$ subjects in the longitudinal case, with margin $\sum_j b_j$.

Hence, maximizing $m(a, d)$ is also motivated by a learning-theoretic perspective. Furthermore, this bound gives useful insight in the algorithm. Firstly, if the average margin cannot be made strictly positive, the algorithm doesn’t learn from...
the \( m \) (independently sampled) subjects. Secondly, the average margin has to grow faster than \( n \) in order for new time-samples \( t_j \) to be useful. Thirdly, the dimensionality \( p \) of the problem does not influence its learning behaviour directly. This is an important insight as it says that this approach is not hampered directly by the curse of dimensionality. Note however that one has to be careful not to have \( r \) grow unboundedly when having the dimension increase.

IV. CASE STUDIES

A. A simple case

The above method is applied to simple simulations where \( p = 36 \). This experiment abstracts the task of extracting relevant features of images (scans) of the brains of evolving species, and how to separate between two different evolution strategies. Think of one subject \( X_i \) in the longitudinal setting as a single (evolutionary) breeding line. That is, initially all individuals are sampled from a global pool. We distinguish between two classes: the breeding lines subject to artificially selection, and the control lines. Each new individual is a slight mutation of its direct ancestors. But lines of the first class have different selection strategies than the lines of the control class. An example of each is given in Fig. 1.

At \( t = 0 \), \( X_i(0) \sim \mathcal{N}(0_p, I) \) with \( I \) the identity matrix of appropriate dimension. The mechanism of control is implemented by the following transition rule

\[
X_i(t+1) = X_i(t) + \sigma U_{i,t}, \quad U_{i,t} \sim \mathcal{N}(0_p, I),
\]

with parameter \( \sigma \) indicates the level of the noise. Artificial selection in turn implements the following rule

\[
X_i(t+1) = X_i(t) + \sigma U_{i,t} + e_{i,t}, \quad U_{i,t} \sim \mathcal{N}(0_p, I),
\]

where \( e = (2, 2, 0_{p-2})^T \in \mathbb{R}^p \) denotes the drift. The vector \( w_0 \propto (1, 1, 0_{p-2})^T \in \mathbb{R}^p \) as shown in Fig. 2(a) represents the true preference in the artificial selection protocol.

![Fig. 1. Two example lines of the simple example with \( X_i(t) \in \mathbb{R}^{36} \) for \( t = 1, 5, 10 \). Left: a breeding line \( X_i \) (‘artificial selection’) evolving as in eq. (19). Right: a breeding line \( X_i \) (‘control’) evolving as in eq. (18). Here, \( \sigma = 0.5 \).
](image)

We applied the LSVM method to this simple case. Here, \( C \) is fixed to 0.001 according to the training validation performance. By solving the dual form of eq. (6), one obtains the matrix \( W \) as shown in Fig. 2(b), which is comparable to the ‘true’ preference \( w_0 \) in Fig. 2(a).

![Fig. 2. Representation of the recovered result. (a) ‘true’ preference \( w_0 \); (b) values of \( w \) as recovered by by solving the dual form of eq. (6); (c) P-value for each pixel by permutation test; (d) values for thresholding the adjusted P-value by the multiple testing method of BH. Here, \( C = 0.001, \sigma = 0.5, 10 \) replicates for each treatment, \( t = 10 \) for each replicate, and using 10000 permutations for the test.
](image)

B. Significance test

Generally, a significant results is accepted when a P-value is less than a pre-fixed significance level \( \alpha \) (say 0.05) [18], [19]. The P-value is the probability of attaining results that are described by the null hypothesis \( H_0 \), while \( \alpha \) is the probability of rejecting \( H_0 \) given that it holds true. Here, \( H_0 \) denotes for each pixel it is irrelevant (or that \( w_i = 0 \)). Practically, the permutation test [20] is used to check if the inferred \( \hat{w} \) by LSVM is significant. This technique is widely used in medical image processing [21], [22]. For images with high dimension, the inference of \( \hat{w} \) is an instance of a so-called multiple testing problem [23]. One infers all elements of \( \hat{w} \) simultaneously for all \( p \) pixels. Several techniques have been developed to correct the testing [24], [25]. Here, corrected P-values for each pixel \( i \) are computed as in [26], referred to as BH.

For the simple case, the heat-map of P-values for each individual is presented in Fig. 2(c). The P-values of the first two pixels are 0.0007 and 0.0001 respectively, which are much smaller than that of the others. To avoid rejecting the \( H_0 \) for some pixels wrongly, the adjusted P-values are calculated according to the BH method. For the first two relevant pixels, they are 0.013 and 0.0036 respectively, as shown in black in Fig. 2(d). That for other irrelevant ones are around 1, as shown in white in Fig. 2(d).

C. Artificial fish brain experiment

A more complicated case with a so-called artificial fish brain is studied. An example of the brain image is shown in Fig. 3(a). The resulting images have a value of \( p = 64 \times 64 \times 64 \), and are arranged in a 3D structure as depicted in Fig. 3. The region of interesting (ROI) is colored in yellow. It corresponds to the telencephalon part of real fish brain which takes the charge of movements [27]. In the simulation, the size of telencephalon part \( r \) changes over generations \( t \) along the Z-axis. The 4th line has a Telencephalon of size \( \max(0, s_t) \). The transition rules for artificial selection and control are as follows:

\[
\begin{align*}
{s}_{i,t+1} &= s_{i,t} + a_0 + \lambda U_{i,t}, & \text{artificial selection} \\
{s}_{i,t+1} &= s_{i,t} + \lambda U_{i,t}, & \text{control},
\end{align*}
\]

By solving the dual form of eq. (6), one obtains the matrix \( w \) as shown in Fig. 2(b), which is comparable to the ‘true’ preference \( w_0 \) in Fig. 2(a).
where $U_{i,t} \sim \mathcal{N}(0,1)$, $a_0 > 0$ the drift term in the artificial selection and $\lambda$ the fluctuation level of $s_i$ in both cases over generations.

From Fig. 3, one finds that LSVM detected the consistently changing part of the brain (the ‘telencephalon’). The map of P-value for each voxel of this case is presented in Fig. 3(b). The P-values for the relevant part are close to 0 while that for the other unchanged parts are almost 1, which should be shown in white color. However, for the sake of illustration, these are omitted from the 3D plot.

**D. Discussion: performance of the LSVM**

The efficacy of the LSVM is compared with the other three ones: Binary Support Vector Machine (SVM), Linear Discriminant Analysis (LDA) and functional linear discriminant analysis (FLDA) as described in [9]. Performance of those 4 are compared using the classical Shepp-Logan (SL) phantom dataset, $p = 64 \times 64$, as displayed in the inset of Fig. 4). As before in eq. (20), the size of the ellipsoid (short axis, left black ellipsoid) is varied according to a simple rule. As presented in the main panel of Fig. 4, the LSVM and FLDA obviously work better than their original versions. Application of the sign-rank test for testing the difference between LSVM and FLDA gives a P-value of 0.3%, indicating that LSVM performs significantly better than FLDA. Since (F)LDA requires computation of the inverse of a $p \times p$-matrix, it is not suited well to handling large-dimensional tasks.

**V. CONCLUSION**

In this paper, we extend the classical SVM algorithm to classify the longitudinal data. Instead of concatenating all images of a breeding line together, we introduced the parameterized margin. As a direct consequence, irregular sampling can be handled straightforwardly. The approach is tailored to the evolutionary setting where the breeding lines originate from a common pool. The derivation results in a convex program which can be solved efficiently, while the dual derivation is used to handle high-dimensional problems proficiently. Finally, we indicate a technique (‘permutation test, and corrections using BH) to convert the inference into P-values which are useful for statistical analysis. Numerical studies indicate the efficiency and usefulness of this approach in an artificial evolutionary experiment.

Two important issues are pending: (i) while we can handle large dimensions of $p = O(10^5)$, even larger dimensionalities are encountered in brain imaging. This points to the use of methods of wavelets or related. (ii) correct tuning of the value of $C$ is crucial for obtaining the desired performance. It is a challenging question how to do this effectively in case no (artificial) validation sets are available.

**ACKNOWLEDGMENT**

This research was funded with the grant 2013.0072 of the “Knut and Alice Wallenberg Foundation” for research on “Linking Social behaviour to the brain”.

**APPENDIX A: DUAL FORM OF EQ. 4**

The dual form of eq. 4 can be derived by a variable transform as is done for the derivation of the standard SVM [5]. We want to choose the new variables $v, a’, b’, d’$ such that

$$\sum_{i,j} (a_{ij} + d) \frac{\|v\|_2}{\sum_{i} n_i} = 1.$$ (21)

Hence, maximizing the total margin $\left(\sum_{ij} (a_{ij} + d)\right)$ becomes equivalent to minimizing the norm $\|v\|_2$. Then, the classifier becomes

$$\arg \min_{v,a’} \frac{1}{2}\|v\|_2^2$$ s.t. $$y_i (v^T X_i(t_{ij}) + b) \geq 1 + a’ \left(t_{ij} - \frac{\sum_{k} (t_{ki})}{\sum_{k} n_k}\right) \quad \forall i = 1, \ldots, n, \forall j = 1, \ldots, n_i,$$ (22)

where

$$\begin{cases} a = a’/\|v\|_2 \\ d = \left(1 - \frac{\sum_{i,j} a’ t_{ij}}{\sum_{i} n_i}\right)/\|v\|_2 \\ b = b’/\|v\|_2 \\ w = v/\|v\|_2. \end{cases}$$ (23)
Equation (22) is known as a quadratic programming problem with objective function expressed as a quadratic function of variable w and linear inequality constraints. It can be solved by standard optimization packages. However, we are more interested in the dual form for the longitudinal data set as it has higher efficiency in coping with large data sets by taking the advantage of the powerful kernel trick.

The idea is to consider the Lagrangian form of eq. (22) by introducing the Lagrangian multipliers (also known as dual variables). Thus, we take into account the advantage of the powerful kernel trick. The dual form for the longitudinal data set as it can be solved by standard optimization packages. However, we are more interested in the dual form for the longitudinal data set as it is straightforward to find the dual form of eq. (6) for the noise case. The original primal parameters w, a, b and d are obtained by normalizing v, a', b' and d' with respect to the norm of v. It is straightforward to find the dual form of eq. (6) for the noise case. The final formula are similar to eq. (28) while the tuning parameter C appearing as an additional constraint on the Lagrangian multipliers αij as:

\[ 0 \leq \alpha_{ij} \leq C, \quad \forall \ i, j. \]

**References**


