TIME SERIES CLASSIFICATION WITH SUPPORT VECTOR MACHINE AND DATA AUGMENTATION

By

Esma Birisci,

Dr. Sounak Chakraborty, Project Supervisor

A Master Project Submitted to the Department of Statistics,

University of Missouri, in Partial Fulfillment of the Requirements of the Degree

Master of Arts

Columbia, Missouri

December, 2011

Acknowledgement

I would like to thank Dr. Sounak Chakraborty for his consistent support and encouragement in the past two year. His initial ideas, insightful suggestions, and wise management have made the completion of this work possible. I have learned a lot from working with him, his active attitude towards research. I am grateful to my committee members, Dr. Min Yang, and Dr.Subharub Guha who have provided comments and suggestions to this project.

I am grateful to Republic of Turkey Ministry of National Education for its financial support of my education. I would like to thank University of Missouri- Columbia Statistics Department for providing a great environment for studying and researching.

I am deeply grateful to my husband and my parents, for their constant support, care, and love.

Abstract

In this project I am interesting in two major problems among researches which are time series classification and data augmentation.

The new method of the rectangular representation of the time series called Dynamic Time Warping (DTW) is introduced in the first chapter. This new method of the rectangular representation is based on the notion of warping distance. Dynamic Time Warping (DTW) is a well-known technique to find optimal alignment between two given (time-dependent) sequences. Successful attempts have been made with Support Vector Machine (SVM) in the application of gene expression time series.

In the second chapter, the new algorithm of estimating parameters in missing data and mixture density called EM algorithm. The EM algorithm has ability to handle with missing data and unidentified variables. Therefore it is becoming useful in variety of incomplete-data problem. Successful attempts have been made with EM algorithm in the application of spam dataset.

Table of Contents

Chapter 1 Introduction

1.1. Motivation

Conventional data mining methods are conceived to deal with ‘statics’ databases which does not use ordering records. Although the assumption of order is unnecessary in some applications, there are definitely many other cases where sequential information can significantly enhance our knowledge about the mined data. Adding time dimension to database produces a Time Series and introduced new aspect of data mining.

In the last decade, many researchers have been interesting in the problem of the time series classification. In current research, researchers assumed large amounts of labeled training data set exist. However, in reality obtaining such kind of data can be very hard or expensive. For example, the field of relation marketing in which is classification problems is based on sequential data; the behavior of costumers is observed through the time and their buying attitude to their interactions with company do certainly compose multivariate time series [2].

In the literature time series classification has been studied extensively by machine learning and data mining communities. Several alternative samples for time series classification have been suggested. The most common approach is based on a two-stage procedure [2]. This approach reduced time series classification to a static problem by suitably transforming the set of multivariate input sequences into a fixed number of columns, through different rectangularization mechanisms. Then one of the classification methods is applied for the labeling data. The aim of the rectangularization is reduction the temporal period of the different time series without risk the wealth of information

included in the original data. Dynamic time warping is useful method to do this rectangularization which is based on notion of warping distance, convenient measure of similarity. This distance allows to expose clusters and to predict the class of new temporal sequences.

The task of time series classification is to map each time series to one of the predefined classes for given a set of unlabeled data.

On the other hand sometimes we can come up data which have missing values or incomplete information to the analysis. For example, survival time data always have some missing values because of dead of job transfer. These data might own incomplete but useful information. If we ignore them in the analysis, we can get some biased results. The EM algorithm has ability to handle with missing and unidentified variables, so it is becoming useful in variety of incomplete-data problem. The EM algorithm is a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values.

There are two main applications of the EM algorithm. The first is *E step* which occurs when the data has missing values, computes the expectation of the log-likelihood evaluated using the current estimate for the parameters. The second is *M step* occurs when optimizing the likelihood function is analytically intractable, computes parameters maximizing the expected log-likelihood found on the *E step*. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

1.2. Thesis Outline

The thesis is divided into four chapters. In Chapter 2, the first part of my project provides information about time series classification underlying this research including one of the

rectangularization methods is dynamic time warping. In addition, the experimental results from analyzing gene expression data using support vector machine is given.

In Chapter 3, the second part of my project provides information about EM algorithm which is dealing with missing and unidentified variables. The last chapter summarized the project and discusses future work.

Chapter 2 Time series Classification using Support Vector Machine

2.1 Time Series Classification and Dynamic Time Warping

Usually, a gene expression is analyzed at various time points, and so measured data generates a time series. In order to assess their similarities and differences, one can be compare pairs of time series. However, they could not be compared same times. This has been shown to be very fragile measure that is especially vulnerable to temporal shifts or noise in the series [7]

The alignment of two time series can be easily visualized in *warp space*. Warp space is a two dimension Cartesian coordinate system, in which both dimension are time. One dimension is the time dimension of first series, and the other is the time of second series. It is called first series $\vec{d}$ (for database) and the second series $\vec{q}$(for query).

An *alignment path* or *warping path* is a continuous set of points (x, y) in warp space. They are restricted so that the values in each dimension are monotonically increasing. That is,

$x_i \leq x_{i+1}$ and $y_i \leq y_{i+1}$

The alignment path corresponds to a mapping from one series to the other series. If path contains a point (x, y), then x in $\vec{d}$ maps to y in $\vec{q}$. Small horizontal or vertical sections of the path are analogous to 'gaps' in sequence alignment [18].

In literature have many time series alignment methods to search for path which best aligns the two series, i.e. parameterized time warping, correlation-optimized warping, and segment-based warping. I will explore dynamic time warping in this chapter.

{Ai} are a set of multivariate time series, where each A_i=[ailt] is a rectangular matrix of size LxT_i of real numbers. Here $l \in L$={1,2,…, L }is the index associated to the

attributes of the time series whereas $t \in T_i$={1,2,…, T_i} is the temporal index that may vary in a different range for each A_i.

Every times series also associated with a class label $y_i \in D$. Let H denote set of functions $f: \Re^n \rightarrow D$ that represent hypothetical relationship between A_i and y_i. Time series classification problem consist of defining an appropriate hypotheses space H and a function $f^* \epsilon H$ which optimally describes the relationship between the time series $\{A_i\}$ and their labels $\{y_i\}$ in the sense of minimizing some measure of misclassification. If there are only two classes i.e. D=2 and y_i={-1,1}, we obtained a binary classification problem, for general case is called multicategory classification.

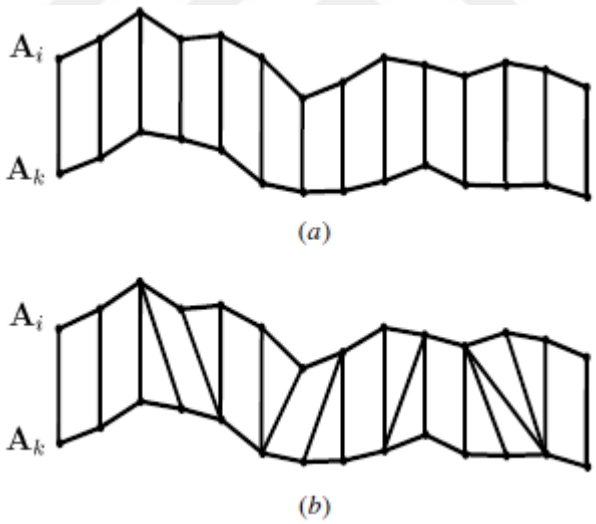

Figure 1: Alignment of Ai and Ak with Euclidean distance (a) and DTW distance (b)

The warping distance between time series A*i* and A*k* is then defined as the length of the shortest warping path in G and provides a measure of the similarity between two temporal sequences which is often more effective than Euclidean matric, as a shown in Figure1.

In order to find optimal alignment between two time series A_i and A_k, G=(V,E) be directed graph whose vertices in V correspond to the pair of time periods (r,s), $r \in T_i$, $s \in T_k$. A vertex v=(r,s) indicates that the r-th value of the time series A_i is matched with the s-th value of A_k. An oriented arc (u,v) connects vertex u=(p,q) to vertex v=(r,s) if and only if one of the following mutually exclusive conditions holds.

$$\{r = p + 1, s = q\}\, V\, \{r = p + 1, s = q + 1\}\, V\, \{r = p, s = q + 1\}$$

The warping distance between A_i and A_k is defined as the length of the shortest warping path in a directed graph;

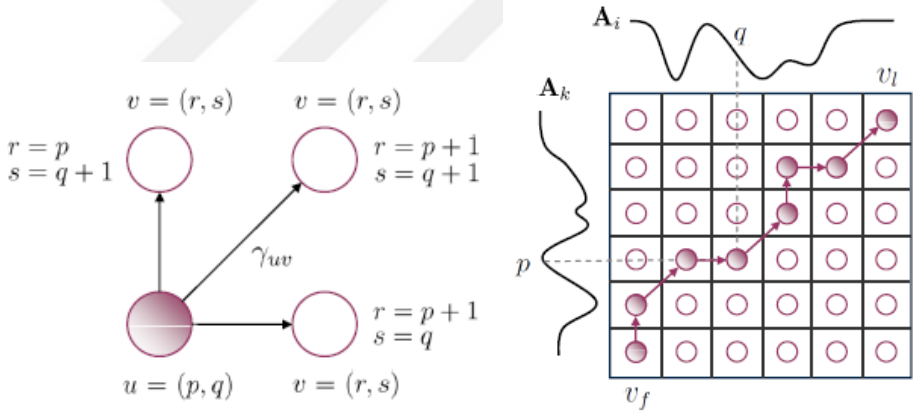

Figure 2: The example of shortest warping path between A_i and A_k in a graph

The arc (u, v) connecting the vertices u=(p,q) and v=(r,s) has length ,

$$\gamma_{u,v} = \sum_{l=1}^{L} (a_{ilr} - a_{kls})^2$$

given A_i by the sum over the attributes of the squared distance associated to the potential alignment of period r in to period s in A_k. A warping path in G is any path connecting the source vertex v_f to the destination vertex v_l.

2.2 Multicatagory SVM in Time Series

The Support vector machine (SVM) is a popular classification paradigm in machine learning and has achieved great success in real applications. However, the standard SVM cannot select variables automatically and therefore its solution typically utilized all the input variables without discrimination. This makes it difficult to identify predictor variables, which is often one of the primary goals in data analysis. In my research, I used SMSVM algorithm which is written by SangJun Lee, Yuwon Kim, Yoonkyung Lee, and Ja-Yong Koo for simultaneous classification.

In the classification problem one is given a training data set on n pairs of covariates and a known class label (xi, yi) for $i = 1 \dots n\ xi\ \in\ R^p$represents the covariates of the ith observation and the response $yi \in \{1 \dots k\}$ is denotes the class that it falls into. In general a classification rule $\emptyset(x)$: $R^p \rightarrow \{1 \dots k\}$ is constructed, based on the training data that generalizes the relationship between xi and class label yi.

The Multicategory SVM proposed by Lee, Lin and Wahba (2004) is a general classification method that extends good theoretical properties of the binary SVM to the multiclass case. It follows the general scheme of finding a k-tuple function $f(x) = f_1(x), \dots, f_k(x)$ which induces a classifier $\emptyset(x) = argmax_{j=1,\dots k} f^j(x)$ via maximum component. For the time series considered in our test set, I may assumed that the input dataset is represented by a *mXn* matrix in which each row is a vector of real numbers $xi\ \in\ R^p$ which represents the corresponding time series ***Ai***. We consider each component $f_j(x)$ as an element of a reproducing kernel Hilbert space (RKHS) $H = \{1\} \oplus \bar{H}$. Then each coordinates $f_j(x)$ can be expressed $b^j + h^j(x)$ with $h^j \in \bar{H}$. A vector value class code *yi* is to be used in place of nominal class label. If $yi = j$, yi=($y_i^1, \dots, y_i^k$) has $y_i^j = 1$ and -1/(k-1) elsewhere. Generally, a regularization method in

aRKHS can be cast as a problem of finding $f(x) = b + h(x) \in H$ in the RKHS that minimizes;

$$\frac{1}{n}\sum_{1}^{n} L(yi, f(xi)) + \lambda\|h\|^2$$

Here $L(yi, f(xi))$ is a loss function measuring goodness of fit, $\|.\|$ is the norm defined on the RKHS $\bar{H}$ and λ is tunable regularization parameter which balances the empirical risk and the penalty associated with f. In this regularization frame work, the MSVM solution $\hat{f}_\lambda$(x)=($\hat{f}_\lambda^1$(x),… , $\hat{f}_\lambda^k$(x)) given λ is defined as the minimize;

$$\frac{1}{n}\sum_{i=1}^{n} L(yi)^t (f(xi) - yi)_+ + \frac{\lambda}{2}\sum_{j=1}^{k}\left\|h^j\right\|^2$$

with the sum to zero constraint $\sum_{j=1}^{k} f^j(x) = 0$ for any $xi \in R^p$. To explain the loss function let $cat(i)$ category of yi and $L_j^{j'}$ be the cost of misclassifying j and j', and define the misclassification cost vector $L(yi) = (L_{cat(i)}^1, \dots, L_{cat(i)}^k)^t$. Then the so-called hinge loss function is $L\big(yi, f(xi)\big) = L(yi)^t (f(xi) - yi)_+$. It can be written explicitly as $\sum_{j=1}^{k} L_{cat(i)}^j \, (f^j(xi) - y_i^j))_+$ where $(x)_+ = \max(x, 0)$.

In my research I used R software to compute a solution to the structured multicategory SVM (SMSVM), which incorporates feature selection with the SVM using the functional ANOVA decomposition. The lasso type L1 penalty on the shrinkage factors of functional components enforces scarcity of the solution. The ordinary MSVM is a special case of the SMSVM when there is no component shrinkage.

In this algorithm we need to specify ANOVA kernel matrixes. These are as below which I used to compare result.

"spline"; for additive models with linear and smooth parts separated.

"spline-t" for additive models with linear and smooth parts combined.

"spline2" for two way interaction models with linear and smooth parts separated.

"spline-t2 " for two way interaction models with linear and smooth parts combined.

2.3 Application in Cell Cycle Data

Experiments were performed on time series gene expression data. The data set is denoted as Yest originally described in [3], contains the genome characterization of the mRNA transcript levels, during the cell cycle of the yeast *Saccharomyces cerevisiae*. In particular measurements were performed at 17 time points with an interval of ten minutes between each pair of records values. The gene expression time series of this data set are known to be associated to five different phases, namely Early G1, Late G1, S, G2, and M, which represent the class values in this setting.

Totally I have 219 different genes expression, each gene expression involves only one of the five class value. Each class has 31, 79, 43, 26, and 30 genes respectively. After making this classification, I use dynamic warping distance function to calculate minimum path for each group. The common path length is twenty nine, so I reveal path direction for each group.

DTW algorithm computes the time axis stretch which optimally maps on the time series (query) onto reference. In this experiment each class reference is their mean vectors. After running DTW algorithm on EarlyG1 class in their path direction, I got a matrix with size 31x29. Thus each group has 29 columns and 31, 79, 43, 26 and 30 rows

what are Early G1, Late G1, S, G2, and M simple size's respectively. In Figure 3 shows each group final path.

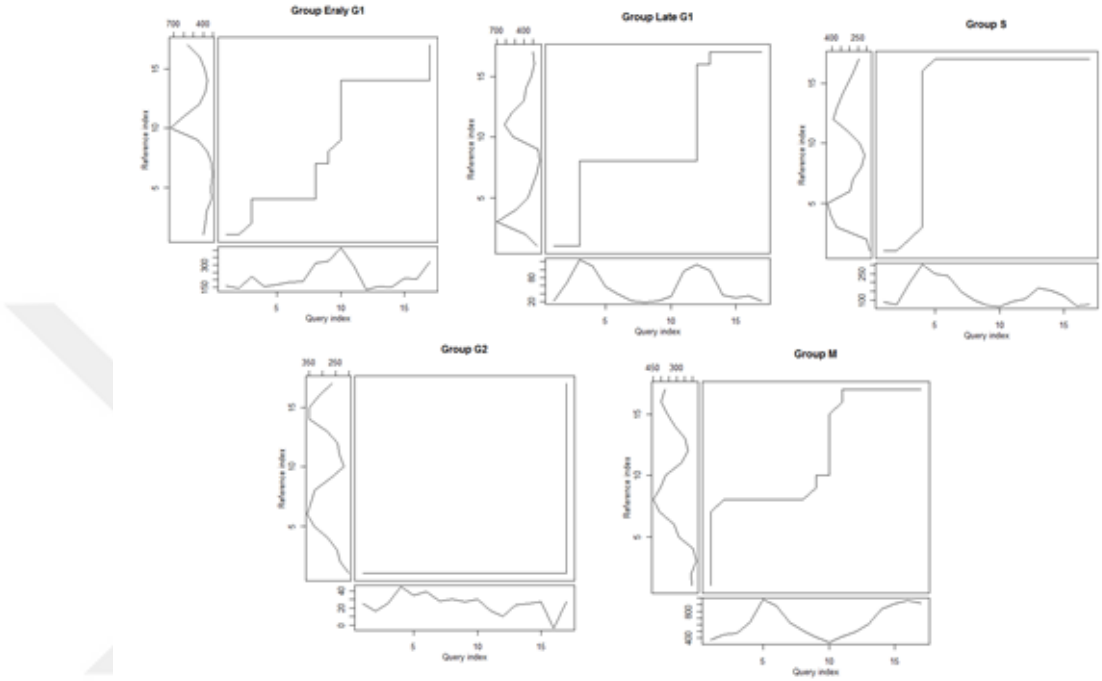

Figure 3: DTW final path for each gene expression time series class

After specify classes and time series length with DTW, I apply SMSVM algorithm to dataset. The result of each kernel matrix is shown in Table 1 which indicates the average error rate obtained by applying smsvm algorithm. The error rate of the competing methods was evaluated by applying 3-fold cross validation each time randomly dividing the dataset into 3 folds for training and testing.

Kernel type	Error Rate
Spline	0.65
Spline-t	0.65
Spline2	0.8166667
Spline -2t	0.3333

Table 1: Average error rate by smsvm

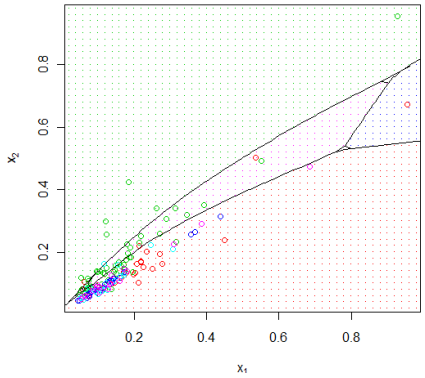

Figure 4: Spline-2t final classification graph

Based on the table 1, the kernel matrix which is spline-2t with parameter 1, gives high accuracy by applying SMSVM algorithm.

Chapter 3 Data Augmentation for Support Vector Machine

3.1 Data Augmentation

Before doing the data analysis it is very important to understand the data structure. Sometimes there may exist of a lot of missing value or incomplete information in the data subject to the analysis. The idea of data augmentation originates in missing value problems. Data augmentation refers to a plan of augmenting the observed data in order to make it easy to analyze [13].

Preferring data augmentation on the massive data set to which SVMs are often applied over the methods is faster. Because of that reason, [14 and 16] has shown that many data augmentation algorithms can be modified to increase their mixing rate. The other advantage of data augmentation methods can be formulated in terms of complete data sufficient statistics, which is considerable advantage when working with large data sets.

In most cases Support Vector Machines (SVM's) binary classifiers are often used with high dimensional covariates. Also as I see before, SVM's typically include a regularization penalty on the vector of coefficients in order to manage the bias-variance trade-off inherent with high dimensional data. In order to avoid getting some biased result we could not ignore some incomplete but useful information. The EM algorithm has capacity to handle with missing data and undefined variables. EM algorithm is a popular tool for solving maximum likelihood problems.

In my project I use the latent variables to implement EM algorithm for obtaining estimates of SVM coefficient. By expressing the support vector optimality criterion as a variance mean mixture of linear models with normal errors, the latent variable representation brings all of conditionally linear model theory to SVM's [16].

The EM algorithms are stable because successive iterations never decrease the objective function. Also EM algorithm exhibits linear (i.e. slow) convergence near the mode.

3.2 Support Vector Machine

The support vector machine classifier is extension of finding linear boundaries in the input feature space where the dimension of the enlarged space is allowed to get very large.

Support vector machines identify a binary outcome $y_i \in \{-1,1\}$ based on a vector of predictors $x_i = (1, x_1, \dots x_{k-1})$. SVM's often include kernel expansions of x_i (e.g. a spline 2t expansion) prior to fitting the model. The L^{α}-norm regularized support vector classifier chosen a set of coefficients β to minimized the objective function

$$d_{\alpha}(\beta, v) = \sum_{i=1}^{n} \max(1 - y_i x_i^T \beta, 0) + v^{-\alpha} \sum_{j=1}^{k} \left|\beta_j / \sigma_j\right|^{\alpha}$$

where $\sigma_j$is the standard deviation of the j[th] element of x and v is a tuning parameter. If hyperplane in x can perfectly separate the sets, $\beta^T x_i \geq 0$, then one classifies observation i as $y_i = 1$, $\beta^T x_i \leq 0$, then one classifies observation i as $y_i = -1$, the result gives separating hyperplane far from any individual observation.

The scaling variable σ_j is the standard deviation the j'th predictor variable with the exception of $\sigma_1 = 1$ for the intercept term. The second term in objective function is a regularization penalty corresponding to a prior distribution $p(\beta|v, \alpha)$.The lasso prior [18,23], corresponding to $\alpha = 1$ is a popular choice because it tends to produce posterior distributions where many of the β coefficients are exactly zero at the mode.

Minimizing equation (1) is equivalent to finding the mode of the pseudo-posterior distribution $p(\beta|\upsilon, \alpha, y)$ defined by

$$p(\beta|\upsilon, \alpha, y) \propto \exp(-d_\alpha(\beta, \upsilon))$$

$$\propto C_\alpha(\upsilon)L(y|\beta)p(\beta|\upsilon, \alpha)$$

The factor of $C_\alpha(\upsilon)$is a pseudo-posterior normalization constant that is absent in the classical analysis. The data dependent factor $L(y|\beta)$ is a pseudo-likelihood

$$L(y|\beta) = \prod_i L_i(y_i|\beta) = \exp\left\{-2\sum_{i=1}^{k} \max(1 - y_i x_i^T \beta, 0)\right\}$$

In principle, one could work with an actual likelihood if each L_i was replaced by the normalized value $\tilde{L}_i = [L_i(y_i)]/[L_i(y_i) + L_i(-y_i)]$, but in the paper they used L_i instead of $\tilde{L}_i$ because it leads to the traditional estimator for support vector machine coefficients.

3.2.1 Mixture Representation

The Pseudo-likelihood contribution $L_i(y_i|\beta)$ is a location-scale mixture of normal. The result allows us to match observation y_i with a latent variable λ_i in such a way that L_i is the marginal from a joint distribution$L_i(y_i, \lambda_i|\beta)$ in which β appears as a part of a quadratics form. This refers that $L_i(y_i, \lambda_i|\beta)$ is a conjugate to a multivariate normal prior distribution.

Theorem 1;

The pseudo- likelihood contribution from observation y_i can be expressed as

$$L_i(y_i|\beta) = \exp\{-2\max(1 - y_i x_i^T \beta, 0)\}$$

$$= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left(-\frac{1}{2}\frac{(1 + \lambda_i - y_i x_i^T \beta)^2}{\lambda_i}\right) d\lambda_i$$

The proofs relies on the integral identity $\int_0^\infty \phi(u|-\lambda,\lambda)d\lambda = e^{-2\max(u,0)}$ where $\phi(u|.,.)$ is the normal density function. The derivation of this identity fallows from Andrews and Mallows (1974), who prove that $\int_0^\infty \frac{\alpha}{\sqrt{2\pi\lambda}} e^{-\frac{1}{2}(a^2\lambda+b^2\lambda^{-1})}d\lambda = e^{-|ab|}$ for any $a, b > 0$. Putting into place, $a = 1, b = u$ and multiplying thought by e^{-u} yields,

$$\int_0^\infty \frac{1}{\sqrt{2\pi\lambda}} \mathrm{e}^{-\frac{u^2}{2\lambda}-u-\frac{1}{2}\lambda}\, d\lambda = e^{-|u|-u}.$$

Finally using the identity $max(u,0) = \frac{1}{2}(|u|+u)$ gives the expression

$$\int_0^\infty \frac{1}{\sqrt{2\pi\lambda}} \mathrm{e}^{-\frac{(u+\lambda)^2}{2\lambda}}\, d\lambda = e^{-2\max(u,0)}$$

which is the asked result.

Theorem 2;

(Pollard, 1946; West, 1987) The prior regularization penalty can be as a scale mixture of normal

$$p(\beta_j|v,\alpha) = \int_0^\infty \phi(\beta_j|0, v^2 w_j \sigma_j^2) p(w_j|\alpha)\ dw_j$$

Where $p(w_j|\alpha) \propto w_j^{-\frac{3}{2}} St_{\frac{\alpha}{2}}^{+}(w_j^{-1})$ and $St_{\frac{\alpha}{2}}^{+}$ is the density function of a positive stable random variable of index $\alpha/2$.

Corollary 1: (*Andrews and Mallows 1974*) The double exponential prior regularization penalty can be expressed as a scale mixture of normal,

$$p(B_j|v,\alpha=1)\int_0^\infty \phi(\beta_j|0, v^2 w_j \sigma_j^2)\frac{1}{2} e^{-\frac{w_j}{2}}\, dw_j$$

and so $p(w_j|\alpha) \sim Exp(2)$ is an exponential with mean 2.

3.2.2 Conditional Distributions

The SVM pseudo-posterior distribution has been shown marginal of higher dimension distribution by theorems 1 and 2. The complete data pseudo-posterior distribution is;

$$p(\beta, \lambda, w|y, v, \alpha)$$

$$\propto \prod_{i=1}^{n} {\lambda_i}^{-1} exp\left(-\frac{1}{2}\sum_{i=1}^{n}\frac{(1+\lambda_i - y_i x_i^T\beta)^2}{\lambda_i}\right)$$

$$\times \prod_{j=1}^{k} {w_j}^{-1/2} exp\left(-\frac{1}{2v^2}\sum_{j=1}^{k}\left(\frac{\beta_j^2}{\sigma_j^2 w_j}\right)\right) p(w_j|\alpha)$$

where in general, $p(w_j|\alpha) \propto {w_j}^{-\frac{3}{2}} St_{\frac{\alpha}{2}}^{+}({w_j}^{-1})$.

This equation is a convenient density with respect to Lebesgue measure on β, λ, w which integrates to a finite number, but it is not a probability density because it does not integrate to one.

Corollary 2; *The full conditional distributions for λ_i is*

$$p({\lambda_i}^{-1}|\beta, y_i) \sim IG(\left|1 - y_i x_i^T\beta\right|^{-1}, 1)$$

Proof; The full conditional distribution is;

$$p(\lambda_i|\beta, y_i) \propto \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{-\frac{(1 - y_i x_i^T\beta - \lambda_i)^2}{2\lambda_i}\right\}$$

$$\propto \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{-\frac{1}{2}\left(\frac{(1 - y_i x_i^T\beta)^2}{\lambda_i} + \lambda_i\right)\right\}$$

$$\propto GIG\left\{\frac{1}{2}, 1, (1 - y_i x_i^T\beta)^2\right\}$$

Equivalently, $p({\lambda_i}^{-1}|\beta, y_i) \sim IG\left(\left|1 - y_i x_i^T\beta\right|^{-1}, 1\right)$ as required.

Corollary 3 *For $\alpha = 1$, the full conditional distribution of w is*

$$p(w_i^{-1}|\beta_j, v) \sim IG\left(v\sigma_j/|\beta_j|, 1\right)$$

Proof: From the integrand in Corollary 1, we have

$$p(w_i|\beta_j, v) \propto \frac{1}{\sqrt{2\pi w_i}} \exp\left\{-\frac{1}{2}\left(\frac{\beta_j^2/v^2\sigma_j^2}{w_i} + w_i\right)\right\}$$

$$\sim GIG\left\{\frac{1}{2}, 1, \beta_j^2/v^2\sigma_j^2\right\}$$

3.3 EM Algorithm

This section shows how the distribution obtained before calculations can be used to build EM-style algorithms to solve for coefficients. Two main applications exist in the EM algorithm. First case consists when the data has missing values because of limitations or problems with the observations process. The second case consists when the likelihood function can be obtained and simplified by assuming that there is additional but missing parameter. The data which is generated by some distribution under assumption with missing values or parameters, X, is incomplete data. The assumption of complete data Z=(X,Y) exist with Y being missing data and that a joint density function also exists as fallows,

$$p(z|\theta) = p(x, y|\theta) = p(y|x, \theta) * p(x|\theta)$$

where θ is a set of unknown parameters from a distribution including a missing parameter.

Now, defining the complete–data likelihood with density function is as below;

$$L(\theta|Z) = L(\theta|X, Y) = p(X, Y|\theta)$$

And original likelihood $L(\theta|X)$ is called incomplete data likelihood function. Since the missing data Y is unknown under a certain distribution, $L(\theta|X, Y)$ can assumed as a function of random variable, Y with constant values X and θ.

$$L(\theta|X,Y) = f_{(X,\theta)}(Y)$$

Using the complete-data log-likelihood function with respect to the missing data Y given the observed data X, the EM algorithm finds its expected value as well as the current parameter estimates at the E step and maximizes the expectation at the M step. By repeating the E and M step, the algorithm is guaranteed to converge to local maximum of the likelihood function with each iteration increasing the log-likelihood.

Expectation (E) Step

First, defining the expectation of the complete-data log likelihood function is as below;

$$Q\left(\theta\middle|\theta^i\right) = E\left[logp(X,Y|\theta)\middle|X,\theta^i\right] \qquad (1)$$

where θ^{i-1} is a set of the current parameters estimates that we use to evaluate the expectation and to increase Q with the new θ for optimization. Here X and θ^{i-1} are known constant and θ is a variable to be adjusted. Since Y is a missing random variable under an assumed distribution, $f\left(y\middle|X\theta^{i-1}\right)$. Then the expectation in the equation can be written as;

$$E\left[logp(X,Y|\theta)\middle|X,\theta^i\right] = \int_{y\epsilon\Omega} \log p\,(X,y|\theta) * f\left(y\middle|X\theta^i\right)dy \qquad (2)$$

where Ω is the space of values where y can take values on and $f(y|X\theta^i)$ is the marginal distribution of the missing data Y depending on observed data and current parameters.

Maximization (M) Step

The M step is maximizing the expectation which obtained in the E step. That is to

$$\theta^{i+1} = argmaxQ(\theta|\theta^i)$$

Maximizing the equation (1) becomes either easy or hard depending on the form of $p(X, y|\theta)$.

Until now, I show basic idea of EM algorithm, now I customized this algorithm into our pseudo-likelihood function which comes from theorem 1.

This part describes an EM algorithm for learning β with fixed value of v.The EM algorithm [20] alternates between an E-step (expectation)and M-step (maximization) defined by;

E-step $\quad Q(\beta|\beta^g) = \int logp(\beta|y, \lambda, w, v)\, p(\lambda, w|\beta^g, y, \lambda) d\lambda\, dw$

M-step $\quad \beta^{(g+1)} = argmax\, Q(\beta|\beta^g)$.

The sequence of parameter values$\beta^{(1)}, \beta^{(2)}$, ...monotically increases the observed data pseudo-posterior distribution: $p(\beta^g|v, \alpha, y) \leq p(\beta^{(g+1)}|v, \alpha, y)$

The Q function in the E- step is the expected value of the complete data log posterior, where the expectation is taken with respect to the posterior distribution evaluated at current parameter estimates. The complete data log-posterior is

$$logp(\beta|y, \lambda, w, v) = c_0(\lambda, w, y, v) - \frac{1}{2}\sum_{i=1}^{n} \frac{\left(1+\lambda_i - y_i x_i^T \beta\right)^2}{\lambda_i} - \frac{1}{2v^2}\sum_{j=1}^{k} \frac{(\beta_j)^2}{\sigma_j^2 w_j}$$

for a suitable constant c_0.

The terms in the first sum are linear function of both λ_i and ${\lambda_i}^{-1}$. However, the λ_i term is free of β, so it can be absorbed in to constant. Thus the relevant portion of equation is a linear function of ${\lambda_i}^{-1}$and ${w_i}^{-1}$, which means that the criterion function $Q(\beta|\beta^g)$ simply replaces ${\lambda_i}^{-1}$and ${w_i}^{-1}$ with their conditional expectations $\hat{\lambda}_i^{-1(g)}$and $\widehat{w}_i^{-1(g)}$ given observed data and current β^g. From **corollary 2** and properties of the inverse Gaussian distribution we have

$$\hat{\lambda}_i^{-1(g)} = E\left(\lambda_i^{-1} \middle| y_i, \beta^{(g)}\right) = |1 - y_i x_i^T \beta^{(g)}|^{-1}$$

The corresponding result for w_j^{-1} depends on α.When α=2 then w_j=1. The general case 0< $\alpha < 2$ is given in the following Corollary of theorem.

Corollary 4. For $\alpha < 2$ if $\beta_j^{(g)} = 0$ then $\widehat{w}_i^{-1(g)} = E\left(w_i^{-1} \middle| \beta^{(g)}, \alpha, y\right) = \infty$. Otherwise

$$\widehat{w}_i^{-1(g)} = \alpha \left|\beta_j^{(g)}\right|^{\alpha-2} (\upsilon\sigma_j)^{2-\alpha}$$

In the case when $\alpha = 1$ we can apply Corollary 3 to obtain

$$\widehat{w}_i^{-1(g)} = \upsilon\sigma_j \left|\beta_j^{(g)}\right|^{-1}$$

which matches the general case. These results lead us to the following algorithm.

Algorithm EM-SVM

Repeat the following until converge

E-Step Given a current estimate $\beta = \beta^{(g)}$, compute

$$\hat{\lambda}_i^{-1(g)} = \left(\left|1 - y_i x_i^T \beta^{(g)}\right|^{-1}\right),$$

$$\widehat{\Lambda}_i^{-1(g)} = diag(\hat{\lambda}_i^{-1(g)}),$$

$$\widehat{\Omega}_i^{-1(g)} = diag(\widehat{w}_i^{-1(g)}),$$

M-step Compute $\beta^{(g+1)}$ as

$$\beta^{(g+1)} = (\upsilon^{-2}\Sigma^{-1}\widehat{\Omega}_i^{-1(g)} + X^T\widehat{\Lambda}_i^{-1(g)}X)^{-1}X^T\left(1 + \hat{\lambda}_i^{-1(g)}\right)$$

The M- step is essentially weighted least square with weights λ_i^{-1}, though it is unusual in the sense that the weights also appear as part of the dependent variable.

Observation i is a support vector if $y_i x_i^T \beta$=1, which means that it lies on the decision boundary.

3.4 Application in Spam Data

The data used for the example is called *spam* implemented in R. The mail spam data described by [21] is a benchmark example in the classification literature. It contains 4601 rows, each corresponding to an email message that has been labeled as a spam or not spam. Each massage is measured on the covariates; there are a total of 58 predictors, including the intercept. I run EM algorithm against this data set several times with several tuning parameter. In our example, I used tuning parameter as -0.5 because beta coefficient converge 0 quite well as Figure 5.

To application EM-SVM algorithm we need initial coefficient for β. Based on this request our initial is a vector of standard normal deviates. To find optimal β coefficient I run algorithm 100 times with different tuning parameter. In figure 1 shows β coefficient with different tuning parameter after 100 iterations.

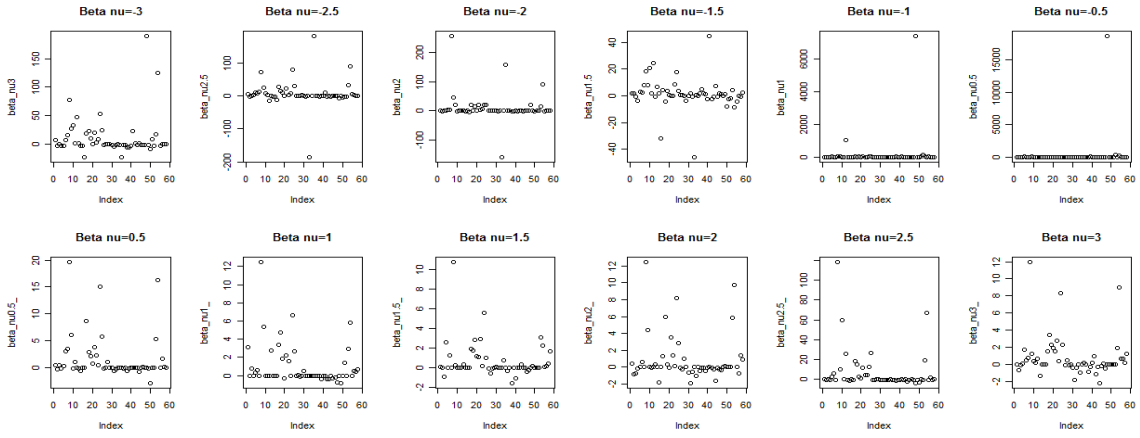

Figure 5: The β coefficients under the α=1 penalty with different tuning parameter

The calculate accuracy of classification, I divided data into two groups, training and test set. I calculate SVM coefficient from the training set and implement these coefficient into pseudo-likelihood equation to get classes -1 or +1.

After this classification I compare the real classes and classes in which the end of the computation and check their accuracy.

Based on this algorithm I got 62% accuracy between real classes label and labeled classes.

Chapter 4 Conclusion and Future Work

First part of this project makes an original contribution to the field of time series classification by developing rectangularization mechanism which is dynamic time warping. This rectangularization mechanism, DTW, was developed based on notion of warping distance. Then, after computing warping distance we applied SVM classification to multidimensional time series. While calculating Dynamic Time Warping distance, each class mean vectors are used as their reference. In the literature DTW gives accurate result with high performance for binary classification, but I show that it also gives well accuracy for multidimensional time series classification by getting 33% error rate.

In future work we will try to improve DTW accuracy by reducing error due to choosing different reference. A possibility to increase the accuracy of DTW is changing the path length for each group. In some applications discontinuities in warp path where the warp path is only increasing one of the time series for a relative long sequence. Modifications that balance the tradeoff between a minimum error warp path and a smooth warp path are important area for future research.

The second of this project, I demonstrate EM algorithm which has very good performance to handle with missing or unidentified values in practice. EM algorithm consists of two processes: The E-step and the M-step. In the E-step the missing data are estimated given the observed data. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. EM algorithm has two important features; It is usually converges in small number of iterations and it is sensitive to initial starting point.

In chapter 3, I utilized EM algorithm to experiment on one data which called ‘spam’, and then I found 62% classification accuracy by using EM-SVM algorithm.

In future study, I may interest finding a procedure of choosing initial values because of its importance to choose appropriate initial values in the EM algorithm. In addition that, we can compare this algorithm with other MCMC algorithm.

References

1. Bilmes, J.A., *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.* International Computer Science Institute, 1998.
2. C. Orsenigo, C.V., *Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification.* 2010.
3. Cho, R.J., Campbell, M.J., Winzeler, E.A.,Landsman, D.,Lockhart, D.j., Davis, R.W., *A genome-wide transcriptional analysis of the mitotic cell cycle.* Molecular Cell2, 1998: p. 65-73.
4. Cui, Y.L.a.Z., *Characterizing the solution path of multicatagory Support Vector Machine.* Statistica Sinica 2006. **16**: p. 391-409.
5. Dellaert, F., *The Expectation Maximization Algorithm.* 2002.
6. Eamonn J. Keogh, C.A.R., *Exact indexing of dynamic time warping.* Knowledge and Information Systems 2004.
7. Eamonn J. Keogh., Michael J. Pazzani, *Derivative Dynamic Time Warping.* SDM conference, Chicago, USA, 2001.
8. Hans, C., *Comment on Article by Polson and Scott.* Bayesian Analysis, 2011. **6**(1): p. 37-42.
9. Hao Helen Zhang, Y.L., Yichao Wu, Ji Zhu, *Variable selection for the multicategory SVM via adaptive sup-norm regularization.* Electronic Journal of Statistics, 2008. **2**: p. 149-167.
10. Hastie, J.F.T. and R. Tibshirani, *Regularization Paths for Generalized Linear Models via Coordinate Descent.* Journal of Statistical Software, 2010. **33**(1).
11. Lifeng Wang, X.S., *On L1-norm multi-class support vector machines: methodology and theory.* Journal of the American Statistical Association, 2007. **102**(478): p. 583-594.
12. Lee, Y.a.C., Z.,, *Characterizing the Solution Path of Multicategory Support Vector Machines* Statistica Sinica, 2006. **16**(2): p. 391-409.

13. Martin A. Tanner, W.H.W., *The Calculation of Posterior Distributions by Data Augmentation.* Journal of the American Statistical Association, 1987. **82**(398): p. 528-540.
14. Meng, X.-L., van Dyk, D. A. , *Seeking efficient data augmentation schemes via conditional and marginal augmentation.* Biometrika, 1999. **86**(2): p. 301-320.
15. Musa H. Asyali, D.C., Omer Demirkaya, Mehmet S. Inan, *Gene Expression Profile Classification: A Review.* Current Bioinformatics, 2006. **1**: p. 55-73.
16. Nicholas G. Polson , S.S., *Data Augmentation for Support Vector Machines.* Bayesian Analysis, 2011. **6**(1): p. 1-24.
17. Niels, R., *Dynamic Time Warping an intuitive way of handwriting recognition* 2004
18. R. Tibshirani. Regression shrinkage and selection via the lasso. Journal ofthe Royal Statistical Society, Series B, 58:267{288, 1996.
19. Richard Durbin; Sean R. Eddy; Anders Krogh & Graeme Mitchison: *Biological sequence analysis* Cambridge University Press,1998
20. Rubin, A.P.D.N.M.L.D.B., *Maximum Likelihood from Incomplete Data via the EM Algorithm.* Journal of the Royal Statistical Society. Series B (Methodological),, 1977. **39**(1): p. 1-38.
21. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009.
22. Xiaopeng Xi, E.K., Christian Shelton, Li Wei, *Fast Time Series Classification using Numerosity Reduction* Proceedings of the Twenty-Third International Conference on Machine Learning, 2006: p. 1033-1040.
23. Zhu,H.Q. *et al*. (2004) Accuracy improvement for identifying translation initiation sites in microbial genomes. *Bioinformatics*, **20**, 3308–3317.