International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume: 3 | Issue: 3 | Mar-Apr 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470

Soft Computing Techniques Based Image Classification using Support Vector Machine Performance

Tarun Jaiswal[1], Dr. S. Jaiswal[1], Dr. Ragini Shukla[2]

[1]Guru Ghasidas Central University, Bilaspur, Chhattisgarh, India
[2]Department of It, CVRU, Bilaspur Kota, Bilaspur, Chhattisgarh, India

How to cite this paper: Tarun Jaiswal | Dr. S. Jaiswal | Dr. Ragini Shukla "Soft Computing Techniques Based Image Classification using Support Vector Machine Performance" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-3, April 2019, pp.1645-1650, URL: https://www.ijtsrd.com/papers/ijtsrd23437.pdf

IJTSRD23437

ABSTRACT

In this paper we compare different kernel had been developed for support vector machine based time series classification. Despite the better presentation of Support Vector Machine (SVM) on many concrete classification problems, the algorithm is not directly applicable to multi-dimensional routes having different measurements. Training support vector machines (SVM) with indefinite kernels has just fascinated consideration in the machine learning public. This is moderately due to the fact that many similarity functions that arise in practice are not symmetric positive semidefinite. In this paper, by spreading the Gaussian RBF kernel by Gaussian elastic metric kernel. Gaussian elastic metric kernel is extended version of Gaussian RBF. The extended version divided in two ways- time wrap distance and its real penalty. Experimental results on 17 datasets, time series data sets show that, in terms of classification accuracy, SVM with Gaussian elastic metric kernel is much superior to other kernels, and the ultramodern similarity measure methods. In this paper we used the indefinite resemblance function or distance directly without any conversion, and, hence, it always treats both training and test examples consistently. Finally, it achieves the highest accuracy of Gaussian elastic metric kernel among all methods that train SVM with kernels i.e. positive semi-definite (PSD) and Non-PSD, with a statistically significant evidence while also retaining sparsity of the support vector set.

KEYWORDS: *SVM, PSD, time series; support vector machine; dynamic time warping; kernel method*

1. INTRODUCTION

We motivated of kernel algorithm because, Firstly, linearity is moderately special, and outside mathematically no model of a real system is actually linear. Secondly, detecting linear relations has been the focus of much research in statistics, soft computing and machine vision for decades and the resulting algorithms are well understood, well developed and efficient. Naturally, one wants the best of both worlds. So, if a problem is non-linear, instead of trying to fit a non-linear model, one can map the problem from the input space to a new (higher-dimensional) space (called the feature space) by doing a nonlinear transformation using suitably chosen basis functions and then use a linear model in the feature space. This is known as the `kernel trick'. The linear model in the feature space corresponds to a non-linear model in the input space. This approach can be used in both classification and deterioration problems. The choice of kernel function is crucial for the success of all kernel algorithms and its variety of types because the kernel establishes preceding knowledge that is available about a task. Accordingly, there is no free dine in kernel choice.

According to *Martin Sewell, 2007*- term kernel is resulting from a word that can be sketched back to c. 1000 and originally meant a seed (contained within a fruit) or the softer (usually edible) part contained within the hard shell of a nut or stone-fruit. The former meaning is now superseded. It was first used in reckoning when it was defined for integral equations in which the kernel is known and the other function(s) unknown, but now has several meanings in mathematics. The machine learning term kernel trick was first used in 1998.

In linear algebra we know that any symmetric matrix K with real valued entries can be written in the form $K = PDP^T$ where $P = (\bar{V}_1, \bar{V}_2, \ldots\ldots, \bar{V}_m), \bar{V}_i$ are eigen vectors of K that form an orthonormal basis (so we also have $P^T = P^{-1}$) and where D is a diagonal matrix with $D_{i,j} = \lambda_i$ being the corresponding eigen values. A square matrix A is positive semi-definite (PSD) i_ for all vectors c we have $c^T Ac = \sum_i \sum_j c_i c_j A_{i,j} \geq 0$. It is well known that a matrix is positive semi-definite iff all the eigen values are non-negative.

In this paper we check the condition of symmetric positive semidefinite with the help of Mercer's Theorem according to the Mercer's Theorem:

The sample $S = x^1, x^2, \ldots, x^m$ includes m examples. The Kernel (Gram) matrix K is an $m \times m$ matrix including inner products between all pairs of examples i.e., $k_{i,j} = k(x^i, x^j)$ is symmetric since $k(x, y) = k(y, x) = \emptyset(x) \cdot \emptyset(y)$

Mercer's Theorem:
A symmetric function $k(.,.)$ is a kernel iff for any finite sample S the kernel matrix for S is positive semi-definite.

One direction of the theorem is easy: if $k()$ is a kernel, and K is the kernel matrix with $K_{i,j} = k(x_i, x_j)$. Then

$$c^T Kc = \sum_i \sum_j c_i c_j K_{i,j} = \sum_i \sum_j c_i c_j \emptyset(x_i)\emptyset(x_j) =$$
$$(\sum_i c_i \emptyset(x_i))(\sum_i c_j \emptyset(x_j)) = \left\|(\sum_j c_j \emptyset(x_j))\right\|^2 \geq 0.$$

Theorem:
Consider a finite input space $S = x^1, x^2, \ldots, x^m$ and the kernel matrix K over the entire space. If K is positive semi-definite then $k(.,.)$ is a kernel function.

Proof: By the linear algebra facts above we can write $K = PDP^T$.

Define a feature mapping into a m-dimensional space where the lth bit in feature expansion for the other direction we will prove a weaker result.
Example x^i is $\emptyset l(x^i) = \sqrt{\lambda_i}\,(\bar{\lambda}_i)_i$.
The inner product is

$$\emptyset(\vec{x}^i) \cdot \emptyset(\vec{y}^j) = \sum_{l=1}^{m} \emptyset l(x^i)\emptyset l(x^j)$$
$$= \sum_{l=1}^{m} \lambda_l (V_i)_l\,(V_i)_l$$

We want to show that

$$k(x^i, x^j) = \emptyset(\vec{x}^i) \cdot \emptyset(\vec{y}^j)$$

Consider $i, j\ th$ entry of the matrix $K = k(x^i, x^j)$. We have the following identities where the last one proves the result.

$$K_{i,j} = [PDP^T]_{i,j}$$
$$= \left|[PD]P^T\right|_{i,j}$$
$$[PD] = (\bar{V}_1, \bar{V}_2, \ldots, \bar{V}_m)D$$
$$[PD]_{i,j} = (V_i)_i \lambda_i$$
$$\left[[PD]P^T\right]_{i,j} = \sum_{l=1}^{m} (V_i)_l \lambda_l\,(V_i)_l$$

Note that Mercer's theorem allows us to work with a kernel function without knowing which feature map it corresponds to or its relevance to the learning problem. This has often been used in practical applications.

In real-life solicitations, however, many similarity functions exist that are either indefinite or for which the Mercer condition is difficult to verify. For example, one can incorporate the longest common subsequence in defining distance between genetic sequences, use BLAST similarity score between protein sequences, use set operations such as union/intersection in defining similarity between transactions, use human-judged similarities between concepts and words, use the symmetrized Kullback-Leibler divergence between probability distributions, use dynamic time warping for time series, or use the refraction distance and shape matching distance in computer vision [1,2,3,4]. Outspreading SVM to indefinite kernels will greatly expand its applicability. Recent work on training SVM with indefinite kernels has generally warped into three categories: Positive semidefinite (PSD) kernel approximation, non-convex optimization (NCO) and learning in Krein spaces (LKS). In the first approach, the kernel matrix of training examples is altered so that it becomes PSD. The motivation behind such approach is to assume that negative eigenvalues are caused by noise [5,6]. The concluding approach was introduced by Luss and d'Aspremont in 2007 with enhancements in training time reported [7,8,9]. All the kernel approximation methods above guarantee that the optimization problem remains convex during training. During experiment, however, the original indefinite kernel function is used. Hence, training and test examples are treated contradictorily. In addition, such methods are only useful when the similarity matrix is approximable by a PSD matrix. For other similarity functions such as the sigmoid kernel that can occasionally yield a negative semidefinite matrix for certain values of its hyper-parameters, the kernel approximation approach cannot be utilized.

In the second approach, non-convex optimization methods are used. SMO type decomposition might be used in finding a local minimum with indefinite similarity functions [10]. Haasdonk interprets this as a method of minimizing the distance between reduced convex hulls in a pseudo-Euclidean space [4]. However, because such approach can terminate at a local minimum, it does not assurance learning [1]. Similar to the previous approach, this method only works well if the similarity matrix is nearly PSD.

The next approach that has been proposed in the writings is to extend SVM into the Krein spaces, in which a reproducing kernel is decomposed into the sum of one positive semidefinite kernel and one negative semidefinite kernel [11,12]. Instead of minimizing regularized risk, the objective function is now stabilized. One fairly recent algorithm that has been proposed to solve the stabilization problem is called Eigen-decomposition SVM (ESVM) [12]. While this algorithm has been shown to outperform all previous methods, its primary drawback is that it does not produce sparse solutions, hence the entire list of training examples are often needed during prediction.

The main contribution of this paper is to establish both theoretically and experimentally that the 1-norm SVM [13], which was proposed more than 10 years ago, is a better solution for extending SVM to indefinite kernels. More specifically, 1-norm SVM can be interpreted as a structural risk minimization method that seeks a decision boundary with large similarity margin in the original space. It uses a linear algebra preparation that remains convex even if the kernel matrix is indefinite, and hence can always be solved quite efficiently. It uses the indefinite similarity function (or distance) directly without any transformation, and, hence, it always treats both training and test examples consistently. In addition, it achieves the highest accurateness among all the methods that train SVM with indefinite kernels, with a

statistically important indication, while also retaining sparsity of the support vector set. In the literature, 1-norm SVM is often used as an surrounded feature selection method, where learning and feature selection are performed concurrently [14, 13, 15, 17, 16,18]. It was studied in [13], where it was argued that 1-norm SVM has an advantage over standard 2-norm SVM when there are redundant noise features. To the knowledge of the authors, the advantage of using 1-norm SVM in handling indefinite kernels has never been established in the writings.

As a state-of-the-art classifier, support vector machine (SVM) has also been examined and applied for time series classification in two modes. On one hand, combined with various feature extraction approaches, SVM can be adopted as a plug-in method in addressing time series classification problems. On the other hand, by designing appropriate kernel functions, SVM can also be performed based on the original time series data. Because of the time axis distortion problem, classical kernel functions, such as Gaussian RBF and polynomial, generally are not suitable for SVM-based time series classification. Motivated by the success of dynamic time wrapping distance, it has been suggested to utilize elastic measure to construct appropriate kernel. Gaussian DTW kernel is then proposed for SVM based time series classification [19, 20]. Counter-examples, however, has been subsequently reported that GDTW kernel usually cannot outclass GRBF kernel in the SVM framework. Lei and Sun [21] proved that GDTW kernel is not positive definite symmetric acceptable by SVM. Experimental results [21, 22] also showed that SVM with GDTW kernel cannot outperform either SVM with GRBF kernel or nearest neighbor classifier with DTW distance. The poor performance of the GDTW kernel may be attributed to that DTW is non-metric. Motivated by recent progress in elastic measure, Zhang et.al propose a new class of elastic kernel it is an allowance to the GRBF kernel [23].There are lots of Advantages of kernel and its types so some of the types we used in this paper for classification [24]:

- The kernel defines a similarity measure between two data points and thus allows one to incorporate prior knowledge of the problem domain.
- Most importantly, the kernel contains all of the information about the relative positions of the inputs in the feature space and the actual learning algorithm is based only on the kernel function and can thus be carried out without explicit use of the feature space. The training data only enter the algorithm through their entries in the kernel matrix (a Gram matrix), and never through their individual attributes. Because one never explicitly has to evaluate the feature map in the high dimensional feature space, the kernel function represents a computational shortcut.
- The number of operations required is not necessarily proportional to the number of features. Support vector machines is one of the most prevalent classification algorithms. It is inspired by deep learning practicalities, which make use of the Vapnik-Chervonenkis dimension to establish the generalization ability of such clan of classifiers [25, 26]. However, SVM has its limitations, which motivated development of numerous variants including the Distance Weighted Discrimination algorithm to deal with the data stacking phenomenon observed in large dimensions [27] and second order conduit programming techniques for handling uncertain or missing values assuming availability of second order moments of data [28]. One fundamental limiting factor in SVM is the need for positive semidefinite kernels.

2. Methods

In standard two-class classification problems, we are given a set of training data $(x_1, y_1) \ldots\ldots (x_n, y_n)$, where the input $x_i \in R^p$, and the output $y_i \in (1, -1)$ is bnary. We wish to find a classification rule from the training data, so that when given a new input x, we can assign a class y from $\{1, -1\}$ to it.

To handle this problem, we consider the 1-norm support vector machine:

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} \left[1 - y_i (\beta_0 + \sum_{j=1}^{q} \beta_j h_j(x_i) \right] \qquad (1)$$

$$s.t. \quad \|\beta\|_1 = |\beta_1| + \cdots \ldots + |\beta_q| \leq s, \qquad (2)$$

Where $D = \{h_1(x), \ldots, h_q(x)\}$ a dictionary of basis functions, and s is a tuning parameter. The solution is denoted as $\hat{\beta}_0(s) and\ \hat{\beta}(s)$; the fitted model is

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{j-1}^{q} \hat{\beta}_j h_j(x) \qquad (3)$$

The classification rule is given by $sign = \hat{f}(x)]$. The 1-norm SVM has been successfully used in classification. We argue in this paper that the 1-norm SVM may have some advantage over the standard 2-norm SVM, especially when there are redundant noise features. To get a good fitted model $\hat{f}(x)$ that performs well on future data, we also need to select an appropriate tuning parameter s. In practice, people usually pre-specify a finite set of values for s that covers a wide range, then either use a separate validation data set or use cross-validation to select a value for s that gives the best performance among the given set.

3. Large similarity margins

Given a similarity function $S(x_i, x_j): x \times x \rightarrow R$ between examples x_i and x_j, we can define similarity between an example x_t and a class $y = l$ to be a weighted sum of similarities with all of its examples. In other words, we may write:

$$s(x_t, l) = \sum_{i=1}^{m} \lambda_i S(x_i, x_t). \mathbb{I}\{y_i = l\} \qquad (4)$$

To denote class similarity between x_i and a class $y = l$. Here, the weight λ_i represents importance of the example x_i to its class y_i. In addition, we can introduce an offset b that quantifies prior preference. Such offset plays a role that is similar to the prior in Bayesian methods, the activation threshold in neural networks, and the offset in SVM. Thus, we consider classification using the rule:

$$\hat{y}_t = \text{sign}\{s(x_t,+1)-s(x_t,-1)+b\}, \qquad (5)$$

Which is identical to the classification rule of 1-norm SVM given in Eq 4. Moreover, we define the similarity margin M_t for example x_t in the usual sense:

$$M_t = s(x_t, y_t) - s(x_t, -y_t) + y_t b \qquad (6)$$

Maximizing the minimum similarity margin can be formulated as a linear program (LP).
First, we write:

Subject to $\max_{\lambda, b, M} M$
$s(x_i, y_i) - s(x_i - y_i) + y_i\, b \geq M, (for\ all\ i)$
$\lambda \geq 0$

However, the decision rule given by Eq. (6) does not change when we multiply the weights **λ** by any fixed positive constant including constants that are arbitrarily large. This is because the decision rule only looks into the sign of its argument. In particular, we can always rescale the weights **λ** to be arbitrarily large, for which $\rightarrow \infty$. This degree of freedom implies that we need to maximize the ratio $M/\|\lambda\|$ instead of maximizing M in absolute terms. Here, any norm $\|\cdot\|$ suffices but the 1-norm is preferred because it produces sparse solutions and because it gives better accuracy in practice.

Since our objective is to maximize the ratio $M/\|\lambda\|$, we can fix M = 1 and minimize $\|\lambda\|$. In addition, to avoid over-fitting outliers or noisy samples and to be able to handle the case of non-separable classes, soft-margin constraints are needed as well. Hence, 1-norm SVM can be interpreted as a method of finding a decision boundary with a large similarity margin in the original space. Such interpretation holds regardless of whether or not the similarity function is PSD. Thus, we expect 1-norm SVM to work well even for indefinite kernels.

Similar to the original SVM, one can interpret 1-norm SVM as a method of striking a balance between estimation bias and variance.

4. Gaussian Elastic Metric Kernel (GEMK)

Before the definition of GEMK, we first introduce the GRBF kernel, one of the most common kernel functions used in SVM classifier. Given two time series *x* and *y* with the same length *n*, the GRBF kernel is defined as where *σ* is the standard deviation.

$$K_{RBF}(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \qquad (7)$$

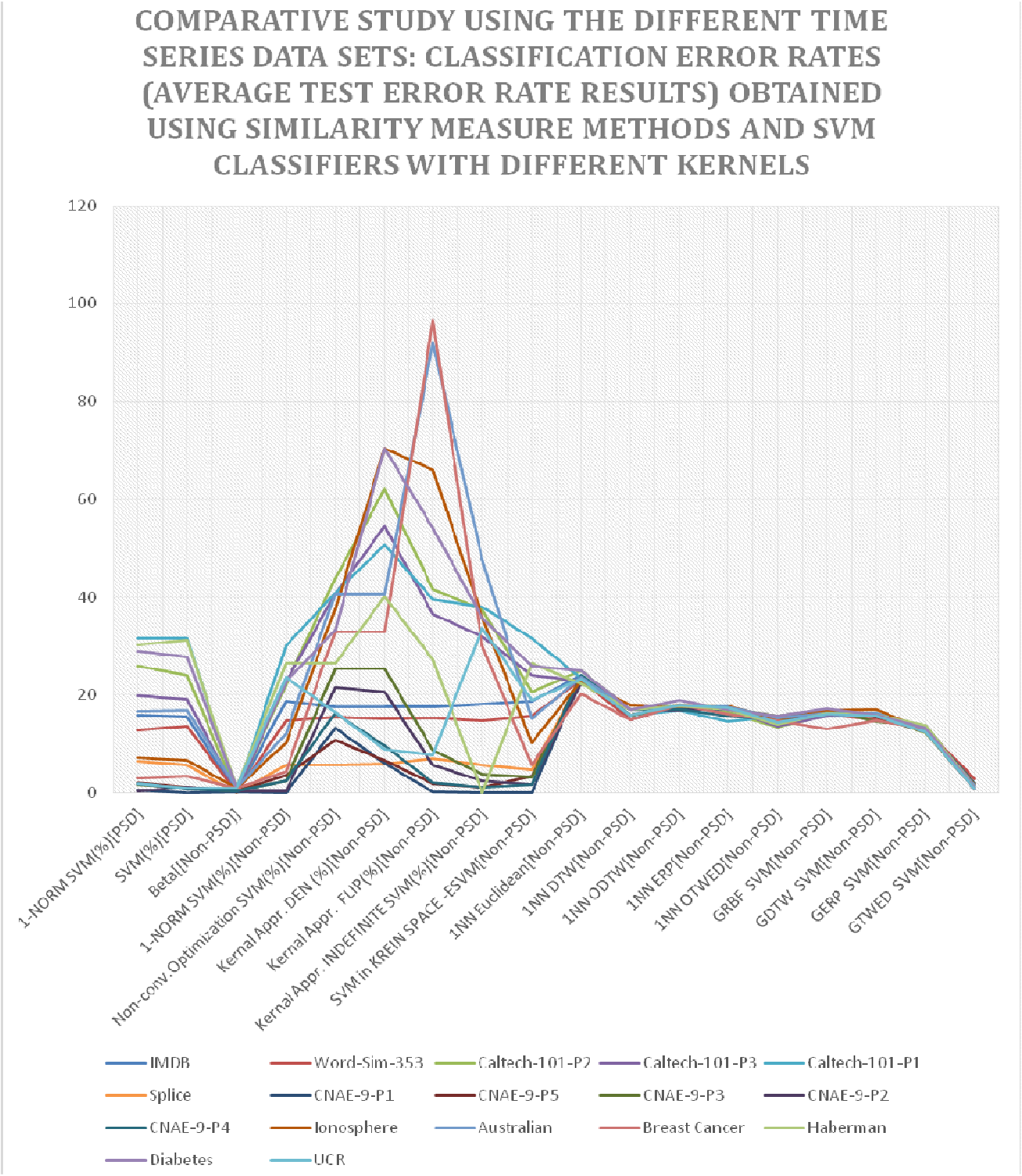
GRBF kernel is a PDS kernel. It can be regard as an embedding of Euclidean distance in the form of Gaussian function. GRBF kernel requires the time series should have the same length and cannot handle the problem of time axis distortion. If the length of two time series is different, re-sampling usually is required to normalize them to the same length before further processing. Thus SVM with GRBF kernel (GRBF-SVM) usually is not suitable for time series classification. Motivated by the effectiveness of elastic measures in handling the time axis distortion, it is interesting to embed elastic distance into SVM-based time series classification. Generally, there are two kinds of elastic distance. One is non-metric elastic distance measure, e.g. DTW, and the other is elastic metric, which is elastic distance satisfying the triangle inequality. Recently, DTW, one state-of-the-art elastic distance, has been proposed to construct the GDTW kernel [19, 20]. Subsequent studies, however, show that SVM with GDTW kernel cannot consistently outperform either GRBF-SVM or 1NN-DTW.

We assume that the poor performance of the GDTW kernel may be attributed to that DTW is non-metric, and suggest extending GRBF kernel using elastic metrics. Thus, we propose a novel class of kernel functions, Gaussian elastic metric kernel (GEMK) functions.

5. Experiments and Results

In this section, we present experimental results of applying different SVM to image classification problems, and determine its efficiency in handling indefinite similarity functions. As shown in last Figure 1, when the similarity function is PSD, performance of Gaussian TWED SVM is comparable to that of SVM. There are different dataset [1, 29-35] we used for measuring the performance. When running statistical significance tests, we find no statistically significant evidence that one method better the other at the 96.45% confidence level. The 1-norm SVM method achieves the highest extrapolative accuracy among all methods that learn with indefinite kernels, while also retaining sparsity of the support vector set other than GTWED SVM. Using the error rate as the performance indicator, we compare the classification performance of Gaussian elastic matching kernel SVM with other different similarity measure methods, including nearest neighbor classifier with Euclidean (1NNED), nearest neighbor classifier with DTW (1NN-DTW) nearest neighbor classifier with ODTW (1NN-ODTW), nearest neighbor classifier with ERP (1NN-ERP) and nearest neighbor classifier with OTWED (1NN-OTWED). Table I lists the classification error rates of these methods on each data set. In our experiments, GRBF-SVM takes the least time among all above kernel methods. Because the complexity of Euclidean distance in GRBF kernel is $O(n)$, while in GDTW, GERP and GTWED, the complexity of DTW, ERP and TWED is $O(n^2)$. Besides, the numbers of support vectors of GERP-SVM and GTWED GTWED-SVM, which are comparable to that of GDTW-SVM, both are more than that of GRBF-SVM. Thus, compared with GRBF-SVM, it also takes more time for GERP-SVM, GTWED-SVM and GDTW-SVM [23].

Figure1:. COMPARATIVE STUDY USING THE DIFFERENT TIME SERIES DATA SETS: CLASSIFICATION ERROR RATES (AVERAGE TEST ERROR RATE RESULTS) OBTAINED USING SIMILARITY MEASURE METHODS AND SVM CLASSIFIERS WITH DIFFERENT KERNELS



6. Conclusion

Widespread research determination has been enthusiastic recently to training support vector machines (SVM) with indefinite kernels. In this paper, we establish theoretically and experimentally that a variant of kernels. We Compare the Study Using the Different Time Series Data Sets: Classification Error Rates (Average Test Error Rate Results) Obtained Using Similarity Measure Methods and SVM Classifiers with Different Kernels. The 1-norm SVM method formulates large-margin separation as a convex linear algebra problem without requiring that the kernel matrix be positive semidefinite. It uses the indefinite similarity function directly without any transformation, and, hence, it always treats both training and test examples consistently. In addition, Gaussian metric kernel methods in the figure achieves the highest accuracy among all methods that train SVM with kernels, with a statistically significant evidence, while also retaining sparsity of the support vector set. This important singularity property ensures that the 1-norm SVM is able to delete many noise features by estimating their coefficients by zero.

References

[1] Yihua Chen, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti. Similaritybased classification: Concepts and algorithms. JMLR, 10:747-776, 2009a.

[2] Gang Wu, Zhihua Zhang, and Edward Y. Chang. An analysis of transformation on nonpositive semidefinite similarity matrix for kernel machines. Technical report, UCSB, 2005.

[3] Yiming Ying, Colin Campbell, and Mark Girolami. Analysis of SVM with indefinite kernels. Advances in NIPS, 22:2205-2213, 2009.

[4] Bernard Haasdonk. Feature space interpretation of svms with indefinite kernels. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27(4):482-492, 2005.

[5] Elzbieta Pekalska, Pavel Paclik, and Robert PW Duin. A generalized kernel approach to dissimilarity-based classification. JMLR, 2:175-211, 2001.

[6] Thore Graepel, Ralf Herbrich, Peter Bollmann-Sdorra, and Klaus Obermayer. Classification on pairwise proximity data. Advances in NIPS, pages 438-444, 1999.

[7] Jianhui Chen and Jieping Ye. Training SVM with indefinite kernels. In Proceedings of ICML, pages 136-143, 2008.

[8] Ronny Luss and Alexandre Aspremont. Support vector machine classification with indefinite kernels. Mathematical Programming Computation, 1(2-3):97-118, 2009.

[9] Yihua Chen, Maya R Gupta, and Benjamin Recht. Learning kernels from indefinite similarities. In Proceedings of ICML, pages 145-152, 2009b.

[10] H.-T. Lin and C.-J. Lin. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, Department of Computer Science, National Taiwan University, 2003.

[11] Cheng Soon Ong, Xavier Mary, Stephane Canu, and Alexander J. Smola. Learning with non-positive kernels. In ICML, 2004.

[12] Gaelle Loosli, Cheng Soon Ong, and Stephane Canu. SVM in Krein spaces. Technical report, 2013.

[13] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. Advances in neural information processing systems (NIPS), 16:49-56, 2004.

[14] Paul S Bradley and Olvi L Mangasarian. Feature selection via concave minimization and support vector machines. In ICML, 1998.

[15] Glenn M Fung and Olvi L Mangasarian. A feature selection Newton method for support vector machine classification. Computational optimization and applications, 28:185-202, 2004.

[16] Melanie Hilario and Alexandros Kalousis. Approaches to dimensionality reduction in proteomic biomarker studies. Briefings in Bioinformatics, 9(2):102-118, 2008.

[17] Hui Zou. An improved 1-norm SVM for simultaneous classification and variable selection. In Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, pages 675-681, 2007

[18] Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao. Feature selection: An ever evolving frontier in data mining. Journal of Machine Learning Research (JMLR) - Work-shop and Conference Proceeding, 10:4-13, 2010.

[19] C. Bahlmann, B. Haasdonk, and H. Burkhardt, "On-line handwriting recognition with support vector machines - a kernel approach,"IWFHR'02, 2002, pp. 49-54.

[20] H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama, "Dynamic time-alignment kernel in support vector machine" NIPS 14, 2002,pp. 921-928.

[21] H. Lei, and B. Sun, "A Study on the Dynamic Time Warping in Kernel Machines," Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, 2007, pp. 839-845.

[22] S. Gudmundsson, T.P. Runarsson, and S. Sigurdsson, "Support vector machines and dynamic time warping for time series," IJCNN'08,2008, pp. 2772-2776.

[23] Dongyu Zhang, Wangmeng Zuo, David Zhang, and Hongzhi Zhang" ,"Time Series Classification Using Support Vector Machine with Gaussian Elastic Metric Kernel", 2010 International Conference on Pattern Recognition,pp.29-32.

[24] Martin Sewell, "lecture notes on kernel",Department of Computer Science University College London, April 2007.

[25] Vladimir N Vapnik. An overview of statistical learning theory. Neural Networks, IEEE Transactions on, 10(5):988{999, 1999.

[26] C. J. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2:121-167, 1998.

[27] J. S. Marron, M. J. Todd, and J. Ahn. Distance-weighted discrimination. Journal of the American Statistical Association, 102(480):1267-1271, 2007.

[28] Pannagadatta K Shivaswamy, Chiranjib Bhattacharyya, and Alexander J Smola. Second order cone programming approaches for handling missing and uncertain data. JMLR, 7: 1283-1314, 2006.

[29] Sofus A. Macskassy and Foster Provost. Classification in networked data: A toolkit and a univariate case study. JMLR, 8:935-983, May 2007.

[30] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. ACM Transactions on Information Systems, 20(1):116{131, January 2002.

[31] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In IEEE CVPR: Workshop on Generative-Model Based Vision, 2004.

[32] M. O. Noordewier, G. G. Towell, and J. W. Shavlik. Training knowledge-based neural networks to recognize genes in DNA sequences. In Advances in NIPS, 1991.

[33] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[34] KP Soman, R Loganathan, and V Ajay. Machine Learning with SVM and other Kernel methods. 2009.

[35] E.J. Keogh, X. Xi, L. Wei, C.A. Ratanamahatana (2006). The UCR Time Series Classification /Clustering, Available at: www.cs.ucr.edu/~eamonn /time_series_data/.