Comparison of Different Kernels using Support Vector Machine

A. Rajeevan and A. Ramanan

Department of Computer Science, University of Jaffna, Sri Lanka.

rajeevan3209@gmail.com, a.ramanan@jfn.ac.lk

Introduction

In general support vector machine (SVM)[1] outperforms other classifiers in its generalisation performance. Kernel[2] methods are becoming increasingly popular for the SVM-based classification tasks. The SVM is well understood when using conditionally positive definite kernel functions. However, in practice, non-conditionally positive definite kernels arise and demand applications in SVM. The procedure of 'plugging' these indefinite kernels in SVM often yields good empirical classification results.

Motivation

In this work we have compared the well known kernels such as linear, RBF, polynomial[2], along with our own implementation of two kernels: $\chi^2$and histogram intersection kernel (HIK)[3] using SVM.

Methodology

Our approach involves …

- We used 10-fold cross-validation to include uncertainties in the performance measures when making comparisons.
- If necessary, the training data were scaled to be in [-1, 1], then the test data were adjusted using the same linear transformation.
- An initial experiment was performed to determine the optimal parameter(s) for each kernel type with a range of values by using a reduced training set that was used by training only on 70% of the training set and validating on the other 30% of the training set.
- In splitting up the training and testing sets, we considered each partition to consist data from every class which is under consideration, so that the classification model is well trained and thereafter tested properly.

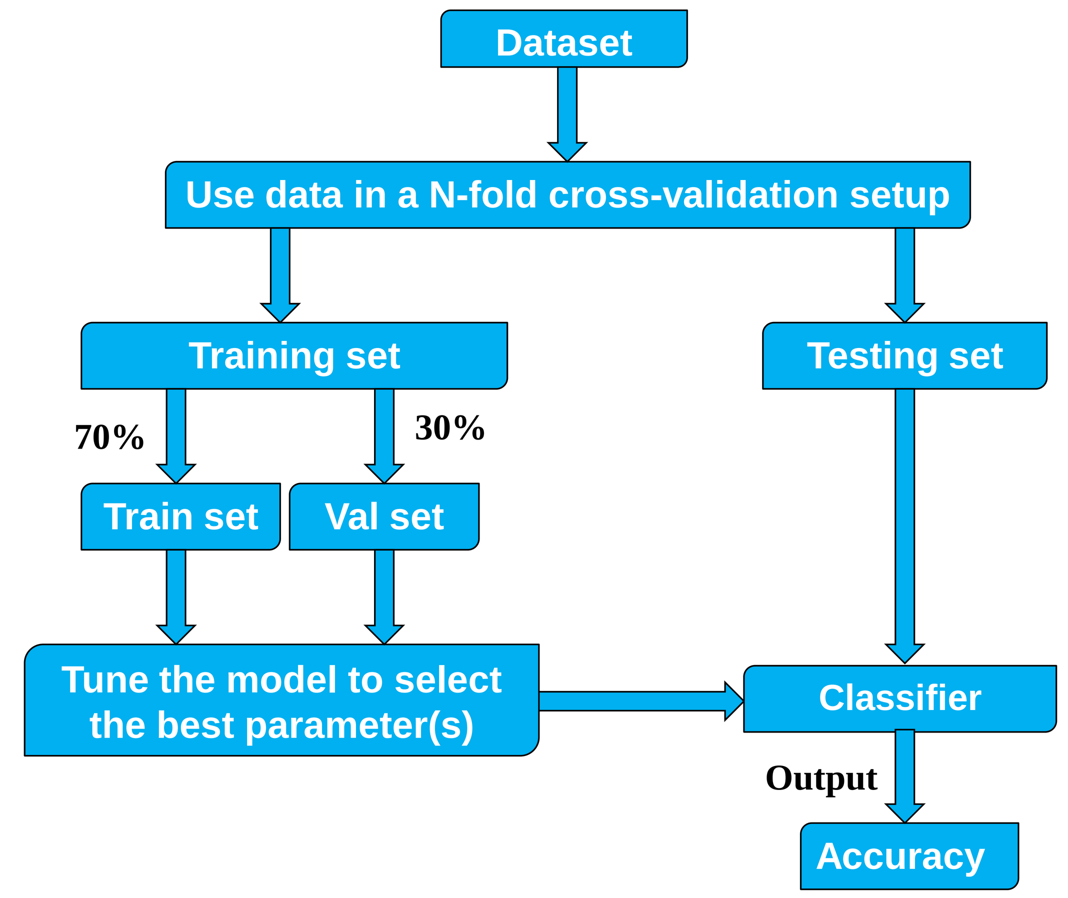

Figure 1: Flow diagram of the proposed method.

Experiments

Support Vector Machines:

A linear support vector machine is composed of a set of given support vectors **z** and a set of weights **w**. The computation for the output of a given SVM with N support vectors $z_1, z_2, \ldots, z_N$ and weights $w_1, w_2, \ldots, w_N$ is then given by:

$$F(x) = \sum_{i=1}^{N} w_i (Z_i, x) + b$$

Kernel Support Vector Machines:

Datasets that are linearly separable as shown in Fig.2 & 3(a) (perhaps with a few exceptions or some noise) are well-handled. But what are we going to do if the data set just doesn't allow classification by a linear classifier?

One way to solve this problem is to map the data on to a higher dimensional space and then to use a linear classifier in that higher dimensional space. SVMs provide an easy and efficient way of doing this mapping to a higher dimensional space, which is referred to as 'the kernel trick'. Let $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$

Then the classifier we have seen so far is:

$$F(x) = \sum_{i=1}^{N} w_i K(Z_i, x) + b$$

K: Kernel function

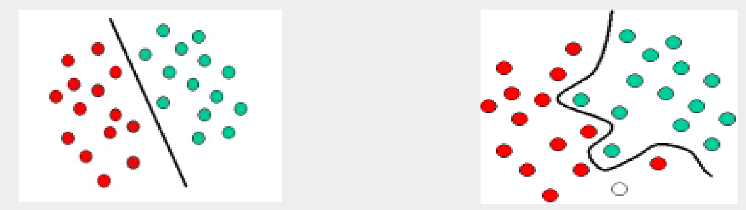

Figure 2: Decision boundary of a binary class (a) linear boundary (b) complex boundary

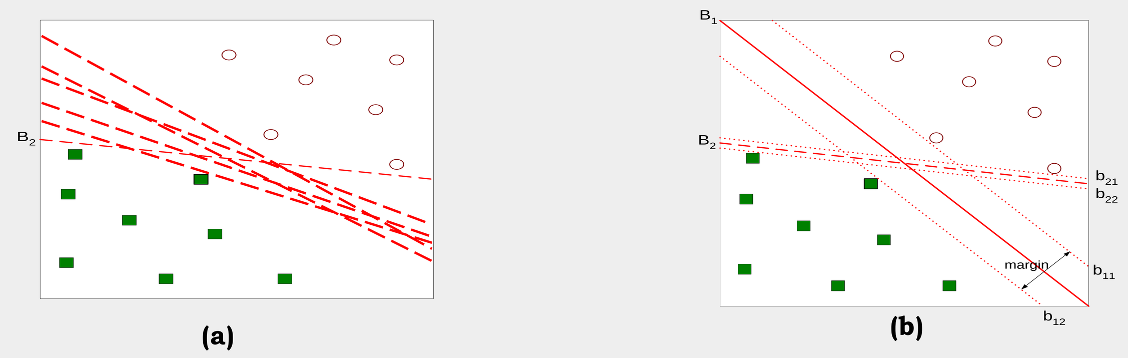

Figure 3: Support vectors (a) all possible SVs (b) maximized margin hyperplane

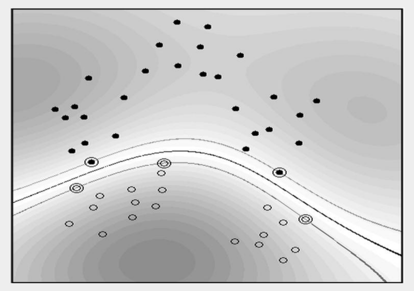

Figure 4: Kernel SVM $K(x, y) = (\alpha x^T y + c)3$

Figure 3(a) shows possible support vectors in a two class problem. Figure 3(b) depicts the maximized margin hyperplane in order to obtain better support vectors. Figure 4 shows a hyperplane which separates better the binary classes by using a kernel function in the feature space.

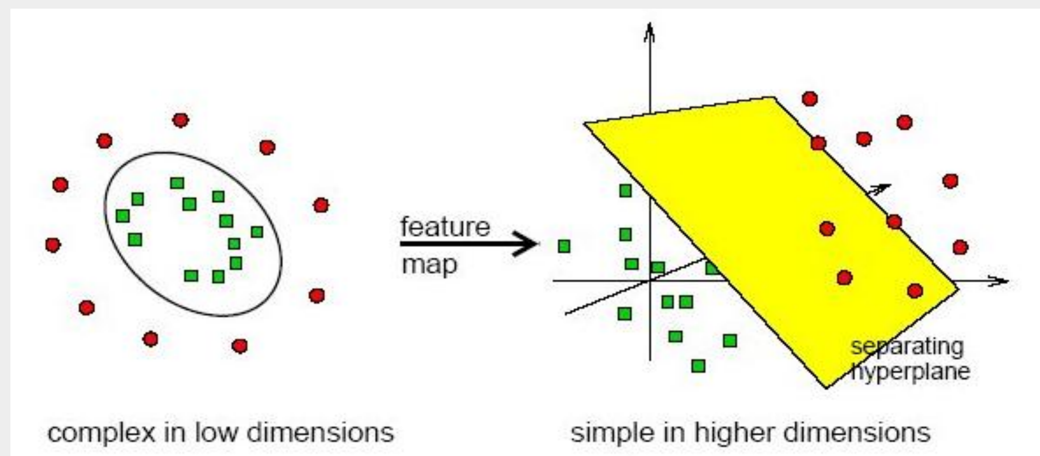

Figure 5: Transformation of data from input space to feature space.

Example kernels

Linear :

$$K(x, y) = x^T y + c$$

RBF:

$$K(x, y) = \exp(-\gamma |x - y|^2)$$

Polynomial:

$$K(x, y) = (\alpha x^T y + c)^d$$

χ^2 - kernel :

$$K(x, y) = \frac{1}{2} \sum_{i=1}^{N} \frac{(x_i - y_i)^2}{(x_i + y_i)}$$

where **x, y** be two vectors, **c** be the penalty parameter, α be the slope, **d** be the degree of the polynomial, and γ be the proportional to the reciprocal of kernel width parameter.

HIK

Let h = (h1, h2... , hd) $\in \mathbb{R}^d_+$ be a histogram, where h could represent the distribution of image features (e.g. SIFT descriptors in a bag-of-keypoints approach). The histogram intersection kernel is defined as shown below.

$$F(x) = \sum_{i=1}^{N} \min(x_i, y_i)$$

Dataset

Table 1: Data statistics of UCI datasets that were used in the experiments

Dataset	#Data	#Attributes	#Classes
iris	150	4	3
wine	178	13	3
Glass	214	9	6
MPEG7_PartB	400	64	20

Results

Table 2: The average accuracy of ten-fold cross-validation for different SVM kernels.

	Kernel				
Dataset	Linear	RBF	Polynomial		
			d = 2	d = 3	d = 4
Iris	96.00 ± 4.63	95.56 ± 3.75	96.67 ± 3.88	96.11 ± 5.27	96.66±3.89
Wine	99.52 ± 1.51	95.95 ± 5.50	99.05 ± 2.01	99.52±1.51	99.05±3.01
Glass	69.75 ± 3.12	68.15 ± 6.91	67.79 ± 5.45	67.39±5.31	67.16±5.28

	Kernel		
Dataset	Linear	HIK	Chi-square
MPEG7_PartB	97±7.33	99.83± 0.53	97.34±3.64

Discussion & Conclusion

- In iris dataset we observed that Polynomial kernel shows better performance than other kernels, but in wine and glass datasets the linear kernel shows better performance as The nature of the data which is scattered in the input space
- In image dataset, the HIK performs much better than linear kernel and chi-squared kernels as the features of images are represented as histograms.

Reference

[1]. Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", 1996.

[2]. Cristianini N and Shawe-Taylor J. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, UK, 2000.

[3]. Subhransu Maji,Alexander C. Berg,Jitendra Malik.Classification using Intersection Kernel Support Vector Machines is Efficient, IEEE Computer Vision and Pattern Recognition ,2008.

[4]. Wu J., & Rehg J. (2009). Beyond the Euclidean dstance: Creating effective visual codebooks using the histogram intersection kernel. In *Proceedings of the IEEE international conference on computer vision (ICCV), 2009.*