Class Information Adapted Kernel for Support Vector Machine

Tasadduq Imam and Kevin Tickle

CINS, CQUniversity, Rockhampton, QLD 4702, Australia.

Abstract. This article presents a support vector machine (SVM) learning approach that adapts class information within the kernel computation. Experiments on fifteen publicly available datasets are conducted and the impact of proposed approach for varied settings are observed. It is noted that the new approach generally improves minority class prediction, depicting it as a well-suited scheme for imbalanced data. However, a SVM based customization is also developed that significantly improves prediction performance in terms of different measures. Overall, the proposed method holds promise with potential for future extensions.

Key words: SVM, Class Informed Kernel, RBF, Sensitivity, Imbalanced data

1 Introduction

Support Vector Machine (SVM) [1, 2] has positioned itself as a state-of-the-art pattern classification technique in many contemporary research areas including brain informatics (e.g. [3, 4]). Given a set of inputs with known class labels (i.e., supervised learning), SVM maps the input space to a high-dimensional feature space such that the training data become linearly separable. The outcome of training SVM is a decision hyperplane that maximizes margin from the class boundaries and, thereby, produces a classifier with high generalization capacity. The explicit mapping from input space to feature space is unknown and is controlled by a function, termed as kernel function, that computes the dot product between the mapped input vectors in the feature space (and dot product is the only processing step, in the unknown feature space, required for SVM training and prediction). While several kernel functions have been proposed and employed in literature, radial basis kernel are often used due to robust performance. These kernel functions can also be viewed as measuring similarity between the feature vectors [5]. The optimization process involved in the SVM training [6, 7] also considers the similarity between feature vectors in its underlying philosophy. However, the similarity (i.e., kernel) is computed based on the input vectors' attributes only and class information of the corresponding vectors are not involved. Viewing kernel value as a similarity measure, this article presents a kernel that takes into consideration the class information of the corresponding vectors. The aim is to conceptualize the impact of including class information during kernel

computation on the classifier's performance. The study reveals that the proposed approach improves the performance of SVM in terms of different measures.

The rest of this article is organized as follows. In Section 2, we present a brief survey on Support Vector Machine classification technique. Section 3 then details our proposed learning approach. A set of experiments, outlining the different characteristics of the proposed approach, and corresponding discussions on outcomes are then highlighted in Section 4. Lastly Section 5 provides a summary of the findings and indicates future potential research.

2 Support Vector Machine

Support Vector Machine is a robust classifier that derives the maximal margin decision hyperplane during training and use it to discriminate test data to one of the two classes (i.e., SVM is a binary classifier working with two class labels only) during prediction [8]. Let the training dataset comprises of tuple $(\mathbf{x}_i, y_i)\ for\, i = 1...N$ where, N is the total number of data, $\mathbf{x}_i$ the i-th attribute vector and y_i (where $y_i \epsilon \{-1, 1\}$) the corresponding class label. Then SVM training can be expressed as the following optimization problem (dual form):

$$min_{\boldsymbol{\alpha}} J_D(\boldsymbol{\alpha}) = \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i.\mathbf{x}_j) - \sum \alpha_i \tag{1}$$

$subject\ to, \sum \alpha_i y_i = 0;\ 0 \leq \alpha_i \leq C\ \ for\, \forall \alpha_i;$

The function $K : \Re x \Re \rightarrow \Re$ is known as the kernel function, that computes the dot product between the data vectors in high dimensional feature space. Several kernel functions have been proposed in literature. Two of these most commonly used kernels are:

- Linear: $(\mathbf{x}_1.\mathbf{x}_2)$
- RBF: $(e^{-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2}$ for $\gamma > 0)$.

The parameter C in the optimization problem (Eq. 1) is a user-defined penalty assignment on training errors. Together, the parameters to kernel (ex., γ for RBF) and C are referred to as the hyper-parameters. SVM training, basically, computes a weight (α_i) associated to each of the data points. In the final solution, data points with $\alpha_i > 0$ are the only important points for classification and are termed as support vectors. SVM training also computes an intercept b for the decision hyperplane. The prediction on a test data $\mathbf{x}$ is given by: $sign\,(\sum_{i=1}^{nsv} \alpha_i^* y_i K(\mathbf{x}_i.\mathbf{x}) + b)$, where nsv is the number of support vectors (i.e., data points with non-zero α).

3 Class Informed Kernel

Noting that dot product relates to cosine similarity, the kernel values (that denotes dot product between vectors in feature space), can be viewed as indicating similarity between the data vectors. But, the calculation of this similarity is

based on the attribute vectors only and does not take into account the respective class labels. Although literature exists that have attempted different kernel modifications to improve prediction performance (e.g. [9]), a kernel that adapts class information in the computation and also addresses the issue that arises during prediction from the use of such kernels (explained in a subsequent paragraph), to the best of our knowledge, is still lacking. Viewing the kernel values as similarity measures, a class informed kernel is as proposed below:

$$K((\mathbf{x}_1, y_1)\ ,\ (\mathbf{x}_2, y_2)) = e^{-\gamma(\|\mathbf{x}_1 - \mathbf{x}_2\|^2\ +\ (y_1 - y_2)^2}\ for\ \gamma > 0; \tag{2}$$

Here, $\mathbf{x}_1, \mathbf{x}_2$ are the input vectors' attributes and $y_1, y_2 \epsilon \{-1\ ,\ +1\}$ are the respective class labels.

The kernel expression in Eq. 2 is similar to the expression of RBF kernel. The difference is the additional term $(y_1 - y_2)^2$. Assuming that class labels are either $+1$ or -1, the value of this additional term results in 0 when both $\mathbf{x}_1, \mathbf{x}_2$ belong to the same class and 4 when $\mathbf{x}_1, \mathbf{x}_2$ belong to the different class. Thus based on the class labels of the vectors for which kernel is computed, an additional weight is added to the expression of RBF. Further, it is to be noted that, the term $\parallel \mathbf{x}_1 - \mathbf{x}_2 \parallel^2$ denotes the distance between the two vectors in input space. For same class input vectors and the proposed kernel K, this distance remains the same as that for RBF kernel. However, for different class input vectors, the addition of the positive weight in effect increases the distance between the vectors (i.e., artificially increases pairwise margin and thus reduces overlap between the different class data).

An issue with use of this class informed kernel lies in the application of it during prediction. While class labels of the support vectors, derived from training SVM, are known, that of the prediction vectors are unknown. To address this issue, we employ a learning framework outlined in Fig. 1. During training, a SVM model is learnt using the proposed class informed kernel. In addition, a second classifier is trained on the training data. This second classifier (termed as *support classifier*) is used to provide an estimate of class label during test phase. The estimated class label coupled with the prediction input vector is then given as input to the trained SVM model and the outcome from the model is the final prediction.

4 Experiment Setups and Results

4.1 Datasets and Software

We perform experiments on five publicly available datasets [10]: diabetes, glass, iris, liver and vehicle. For running SVM, we employ the LibSVM classification technique as implemented in R (through the package kernlab) [11]. Other than the diabetes and liver datasets, the rest of the datasets have originated from multi-class domain. Since SVM is primarily a binary classification technique, we convert the multi-class datasets to binary by considering one of the class as positive class and the rest as negative class. Doing this conversion for each of the

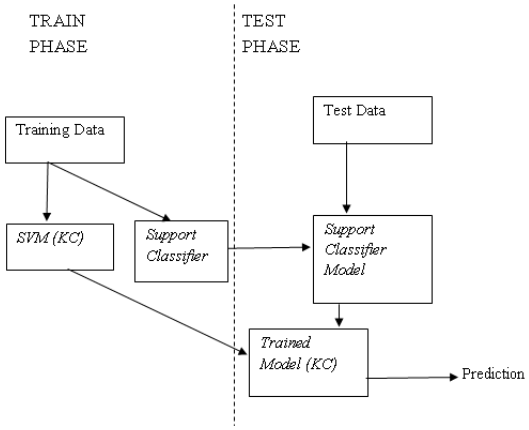


Fig. 1. Framework for learning with class-informed kernel

multiclass data results in total the 15 datasets outlined in Table 1. Table 1 also indicates the total number of data, number of positive and negative class samples and the percentage of the class representation in the datasets. It is noteworthy that some of the datasets are imbalanced (i.e., skewed) in terms of representation of the classes. Imbalanced dataset often arises in many practical applications and it is well known that many classifiers make more prediction errors on minority class samples than that belonging to majority class [12]. Accuracy (ratio of the total number of correctly classified data and the total number of data) is not an appropriate performance measure when datasets is imbalanced. Sensitivity (accuracy for the positive class data) and gmean (geometric mean of the accuracy for positive class data and the accuracy of negative class data) are often employed as performance metric for imbalanced datasets [13, 14]. In our experiments, we focus on all three of these prediction performance measures.

Table 1. Datasets used in the experiments.

Datasets	**Total Data**	**# Positive**	**# Negative**	**% (+)**	**% (-)**
diabetes	768	268	500	34.90	65.10
glass_1	214	70	144	32.71	67.29
glass_2	214	76	138	35.51	64.49
glass_3	214	17	197	7.94	92.06
glass_5	214	13	201	6.07	93.93
glass_6	214	9	205	4.21	95.79
glass_7	214	29	185	13.55	86.45
iris_1	150	50	100	33.33	66.67
iris_2	150	50	100	33.33	66.67
iris_3	150	50	100	33.33	66.67
liver	345	145	200	42.03	57.97
vehicle_1	846	212	634	25.06	74.94
vehicle_2	846	217	629	25.65	74.35
vehicle_3	846	218	628	25.77	74.23
vehicle_4	846	199	647	23.52	76.48

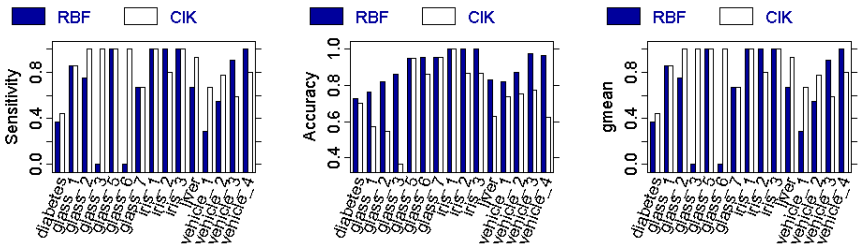


Fig. 2. Performance measures for RBF kernel and CIK (Class informed kernel).

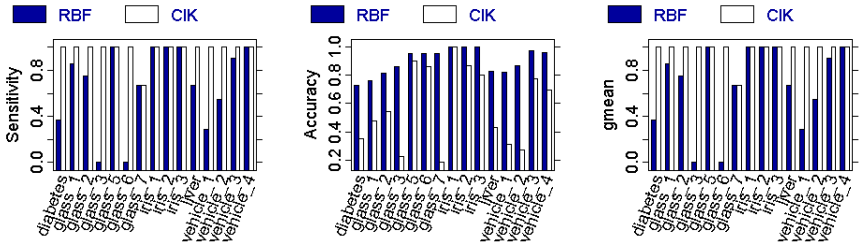
4.2 Experiment with Class Informed Kernel

For our experiments, we randomly split each of the datasets into a train and a test data file. 90% of the total data is used for training and the rest for prediction. Stratified sampling is used to preserve the ratio of positive and negative class data in the train and test files. We analyse the impact of class informed kernel by comparing its performance against a SVM trained on the dataset using RBF kernel. Focus is made on RBF kernel due to its wide popularity and well known robust performance, and also due to the similarity of class informed kernel to the RBF kernel. For RBF kernel, a 10-fold cross validation technique is employed to determine the best parameters (γ and C) for SVM training on the training data file, and the trained model is employed to note prediction performance on test data file. For class informed kernel (CIK), γ and C are set to the best parameter values identified for the RBF kernel. For this initial experiment, we use Naive Bayes classifier (due to its simplicity and high training speed) as the support classifier. Fig. 2 denotes comparison of RBF and CIK. We note that while the CIK (with Naive Bayes as support classifier) based learning does not perform well against the RBF learning in terms of accuracy, in terms of sensitivity the CIK emerges as a clear winner (performs better than or comparable to RBF for 12 datasets out of 15). In terms of gmean, however, there is no clear winner (CIK performs better than or comparable to RBF for 8 datasets out of 15). Thus, we observe that CIK has a positive impact on prediction performance, especially when the dataset is imbalanced (i.e., CIK results in higher prediction of minority class).

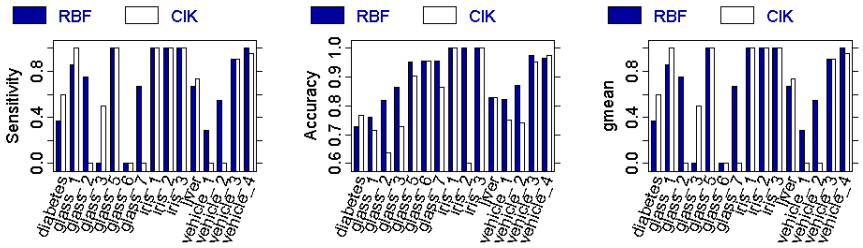
4.3 Varied Support Classifier

In the previous experiment, we have used Naive Bayes as the support classifier. In this section, we present the impact of other different support classifier on prediction outcomes. In particular, we experiment with recursive partitioning and regression trees [15] and a single-hidden-layer neural network. For each of these different support classifiers, prediction performance of CIK is compared against that for RBF. Fig. 3 illustrates the results. We note that the performance for CIK

noticeably varies depending on the support classifier. For recursive partitioning and regression trees (RT), CIK consistently performs better or comparable to RBF for all the datasets in terms of sensitivity. In terms of accuracy and gmean, however, CIK (with RT as support classifier) performs worse than RBF. For single-hidden-layer neural network (NN), CIK performs better than or comparable to RBF for 10 of the 15 datasets in terms of sensitivity. In terms of accuracy, CIK with NN as support performs better than CIK with Naive Bayes as support. However, compared to RBF, CIK with NN performs slightly worse than RBF in terms of accuracy and gmean (CIK with NN is comparable or better than RBF in terms of both accuracy and gmean for 6 of the 15 datasets). Overall, CIK with NN performs better than CIK with other support classifiers focused on so far. In the next subsection, we present CIK with another support classifier that significantly depicts performance improvement over RBF.



(a) RT



(b) NN

Fig. 3. Performance measures for RBF kernel and CIK (Class informed kernel) with (a) recursive regression and partition tree (RT) and (b) single-hidden-layer neural network (NN) as support classifier.

4.4 SVM as Support Classifier

In the previous sub-sections we have experimented with different support classifiers and noted varied effects on prediction performance. More specifically, CIK

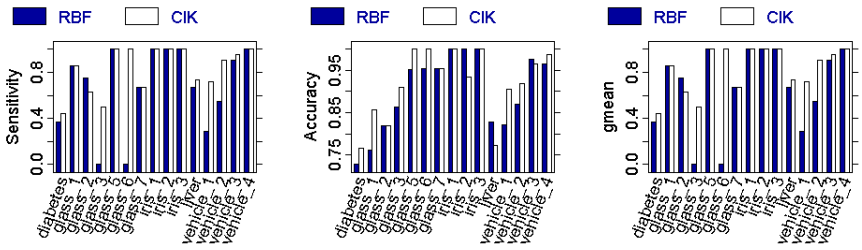


Fig. 4. Performance measures for RBF kernel and CIK (Class informed kernel) with SVM as support classifier.

has generally performed better than RBF in terms of sensitivity, but depicted variations in terms of the other two measures. In this section, we present results for RBF kernel based SVM being used as the support classifier. Thus, for a given train data file, SVM is first trained on the input using the class informed kernel formulation of Eq. 2, and another SVM is trained using RBF kernel. The hyper-parameters are kept at the same values for both of these trainings. During prediction, the RBF based model first predicts the class and the predicted labels along with respective attribute vectors are fed to the CIK based model for prediction. The outcome from CIK is the final prediction. Fig. 4 presents the performance of CIK with SVM as support against that for RBF. We note a significant performance improvement in terms of all the measures. In terms of sensitivity, CIK (with SVM support) performs better or comparable to RBF for 14 of the 15 datasets. In terms of both accuracy and gmean, CIK performs better or comparable to RBF for 12 of the 15 datasets. Thus, not only CIK with SVM improves prediction of minority class (i.e., sensitivity), but also achieves notable prediction improvement for both the classes (as evidenced by improved value of gmean and accuracy). From statistical perspective, we note that the difference in performance for all the three measures are significant using two-tailed sign test [16] with $p < 0.05$. To get further insight on the behaviour of the classifier, we have also recorded the performance of RBF and CIK (with RBF based SVM as support classifier) on the training data in terms of area under ROC (AUC). We note that the CIK performs comparable or better than RBF for 14 of the 15 datasets (with iris_3 being only exception, having slight drop in AUC). We have also noted the ratio of training data incorrectly classified by both RBF and CIK (II), incorrectly classified by RBF but correctly classified by CIK (IC), correctly classified by RBF but incorrectly classified by CIK (CI) and correctly classified by both RBF and CIK (CC). We observe that for majority of the datasets, IC is greater than CI. These findings imply that CIK gains better separability between the class representatives than RBF and which, in turn, provides an explanation of its better prediction performance on the test set in terms of the different measures.

5 Conclusion

This article has presented a new learning approach along with a kernel formulation for SVM incorporating class information. An integral part of this approach is the training of a second (support) classifier and results have been presented for varied support classification schemes. Overall, the proposed kernel based learning (CIK) improves prediction performance in terms of sensitivity and thereby is well suited for imbalanced data classification. Experiments are also conducted using SVM as support classifier and statistically significant prediction performance improvement is noted. The proposed kernel is based on RBF kernel formulation. Future research possibilities lie in the extension of the formulation in terms of other kernels and varied added weights.

Acknowledgments. The authors would like to thank the anonymous reviewers for their insightful recommendations.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (2000)
2. Smola, A.: Advances in Large Margin Classifiers. MIT Press (2000)
3. Xiang, J., Chen, J., Zhou, H., Qin, Y., Li, K., Zhong, N.: Using SVM to Predict High-Level Cognition from fMRI Data: A Case Study of 4* 4 Sudoku Solving. Brain Informatics (2009) 171–181
4. Yang, J., Zhong, N., Liang, P., Wang, J., Yao, Y., Lu, S.: Brain activation detection by neighborhood one-class SVM. Cognitive Systems Research **11**(1) (2010) 16–24
5. Xu, J., Li, H., Zhong, C.: Relevance ranking using kernels. Technical Report MSR-TR-2009-80, Microsoft Research Technical Report (2009)
6. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
7. Fan, R., Chen, P., Lin, C.: Working set selection using second order information for training support vector machines. The Journal of Machine Learning Research **6** (2005) 1918
8. Cristianini, N., Shawe-Taylor, J.: An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge Univ Pr (2000)
9. Wu, G., Chang, E.: KBA: kernel boundary alignment considering imbalanced data distribution. IEEE Transactions on knowledge and data engineering (2005) 786–795
10. : Libsvm data: Classification, regression, and multi-label (2010)
11. Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: kernlab – an S4 package for kernel methods in R. Journal of Statistical Software **11**(9) (2004) 1–20
12. Chawla, N., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter **6**(1) (2004) 1–6
13. Kubat, M., Holte, R., Matwin, S.: Machine Learning for the Detection of Oil Spills in Satellite Radar Images. Machine Learning **30**(2) (1998) 195–215
14. Tan, P., Steinbach, M., Kumar, V.: Introduction to data mining. Pearson Addison Wesley Boston (2006)
15. Breiman, L.: Classification and regression trees. Chapman & Hall/CRC (1984)
16. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7** (2006) 1–30