Investigation of SVMs for multi-class problems

Bernard d'Antras

November 2, 2012

Abstract

Support vector machines are commonly used for binary classification problems, but can also be indirectly applied to multi-class classification problems. One versus one and one versus all technique are quite popular ways to do this. In this paper we show that these methods can beat a k-nearest neighbour classifier in accuracy. A variation on the one versus one method is also described and is shown to be an improvement over the canonical form.

1 Introduction

Support vector machines [2] are one of the most studied classifiers in machine learning, and are considered to be one of the most reliable. But as they are by design only binary classifiers, applying them to multi-class problems is more challenging. This paper will examine two simple algorithms which make use of multiple binary SVMs to solve a multi-class problem. There are other algorithms which also solve this type of problem. These are briefly described here, but have not been implemented due to time constraints.

In this paper we will first describe all the algorithms used in this study, then describe the experiments run on those algorithms to study their performance. Finally the experiment results are presented and discussed.

2 Algorithms

2.1 One versus All SVM

Given n classes, this algorithms consists in training n binary SVM models, using one selected class as the positive class and grouping all other classes in the negative class. Instead of using the predicted label result of the binary SVM, the raw value of the evaluation function is used. This value gives a score to each class, and the class with the highest score is predicted.

An issue with this algorithm is that it is assumed that all binary classifiers are equally reliable, since the score values are not scaled prior to comparison. This may not always be true, for example in a data set consisting of one cluster

of class A and another cluster of classes B and C, the classifier for class A will be much more reliable than the ones for the other two classes.

A possible solution to this problem is to use posterior probabilities from the binary SVM instead of the raw value. The method for obtaining these probabilities is described by Platt [9] and an improved algorithm is described in [11]. This was not implemented in this study due to time constraints.

2.2 One versus One SVM

Given n classes, this algorithms consists in training $\frac{n(n-1)}{2}$ binary SVM models, between each pair of classes. As with the previous algorithm, a score is accumulated for each class based on the results of the binary SVMs. There are various ways to do this, and two were implemented in this study:

- The score for a class is incremented by 1 each time a binary classifier predicts it.
- The score for a class is incremented by the raw SVM value each time a binary classifier predicts it.

This algorithm suffers from the same issue as the previous one: individual binary classifiers may not have the same reliability. Hastie and Tibshirani [6] describe a method of pairwise coupling which uses posterior probabilities of the binary classifiers to determine a class more accurately.

2.3 Other SVM-based algorithms

There exist many other algorithms based on SVMs which solve the multi-class classification problem, but they were not implemented and tested due to time constraints.

Platt, Cristianini and Shawe-taylor [10] propose a method of combining one versus one binary SVMs in a directed acyclic graph.

Dietterich and Bakiri [4] propose a method using error-correcting codes to determine class probabilities from a set of binary SVMs.

Cramer and Singer [3] propose a method which combines all class probabilities into a single SVM-like optimisation problem.

Lee, Lin and Wahba [8] propose another method which extends SVMs to cover multiple classes.

2.4 k-nearest neighbours

The k-nearest neighbours algorithms is not based on SVMs and is instead used as a benchmark to see how well SVM-based algorithms perform compared to an algorithm which explicitly supports multiple classes. The kNN algorithm does not require any training, instead all of the training data is included directly into the model. In order to determine which class a certain example belongs to, the k training data point closest to it (using euclidean distance, but other

Data set	Classes	Features	Total Examples	Examples per class
Seeds	3	7	210	70
Iris	3	4	150	50

Table 1: Statistics on data sets used

distances can also be used) are found and the most common class in that subset is returned.

The main disadvantage of kNN is that it is very slow with large data sets, as each testing examples needs to be compared to all training examples. This is in contrast to SVMs which only store the minimum number of support vectors in the model.

3 Experiments

3.1 Data sets

Two data sets from the UCI collection [5] were used in this study:

- The seeds data set, which consists of geometric features of 3 types of wheat kernel.
- The iris data set, which consists of geometric features of 3 types of iris plant.

All data sets contained real values and were normalised to zero mean and unit variance before use.

Data set statistics are shown in table 1.

3.2 SVM parameter tuning

For all algorithms, the basic binary SVM used was SVM with slack variables and a Gaussian radial basis function kernel. This means that there are two parameters for each SVM: the cost factor C, and the Gaussian kernel width σ. There are two ways to assign these parameters to the models: either each binary model gets its own set of parameters, or the parameters are shared between all binary models in a multi-class model. We decided to follow Hsu and Lin's [7] suggestion to use common parameters within a multi-class model.

Because of the non-linear relationship between these variables and the accuracy of the resulting classifier, an exponential grid search is used to find appropriate values for these variables. For both variables the values $[2^{-4}, 2^{-3}, ..., 2^3, 2^4]$ are tested and the generalisation error is estimated using 5-fold cross-validation.

In case of a tie, we follow the suggestion from Duan and Keerthi [1] to first pick the parameters with the largest σ value, and secondly pick the parameters with the smallest C value.

The results of the grid search are shown table 2.

	1 vs All		1 vs 1 Wins		1 vs 1 Raw	
Data set	C	σ	C	σ	C	σ
Seeds	2	2	4	1	2	0.25
Iris	4	0.0625	0.25	1	0.5	0.5

Table 2: Grid search results for parameter selection

Data set	1 vs All	1 vs 1 Wins	1 vs 1 Raw	kNN
Seeds	10.81 ± 4.23	9.83 ± 4.77	9.10 ± 3.94	11.29 ± 3.67
Iris	2.67 ± 2.68	3.23 ± 2.98	3.10 ± 2.97	3.97 ± 3.06

Table 3: Mean and standard deviation of generalisation error for 20 runs of 5-fold cross-validation (percentages)

For the kNN classifier, a linear search was used and found an optimal value of $k = 5$ for both data sets.

3.3 Performance evaluation

In order to estimate the generalisation performance of the classifiers, 5-fold cross-validation was performed on 20 random permutations of the data set for each classifier.

Results are shown in table 3 and figures 1 and 2.

4 Conclusion

Looking at the results in table 3, we can see that all three SVM-based classifiers generally outperform a k-nearest neighbours classifier. This is slightly surprising as kNN is considered to be a very reliable, although expensive, multi-class classifier.

Another interesting result is that summing up raw scores instead of wins for the one versus one classifier yields slightly better results. This shows that SVM raws values provide a better estimation of classifier reliablity than assuming equal reliability.

Unfortunately the main issue with this study is the small size of the data sets used. The use of small data sets (150 and 210 examples) amplified the noise in the classifier results, for example in the iris data set where the standard deviation of the error rate is as big as the error rate itself. Larger data sets were not used in this study to avoid excessive computation times, but a further study would be able to extract more meaningful results from larger data sets.

A final issue with this study is that both data sets had the same number of classes, which limits the exploration of the effects of class count on classifier

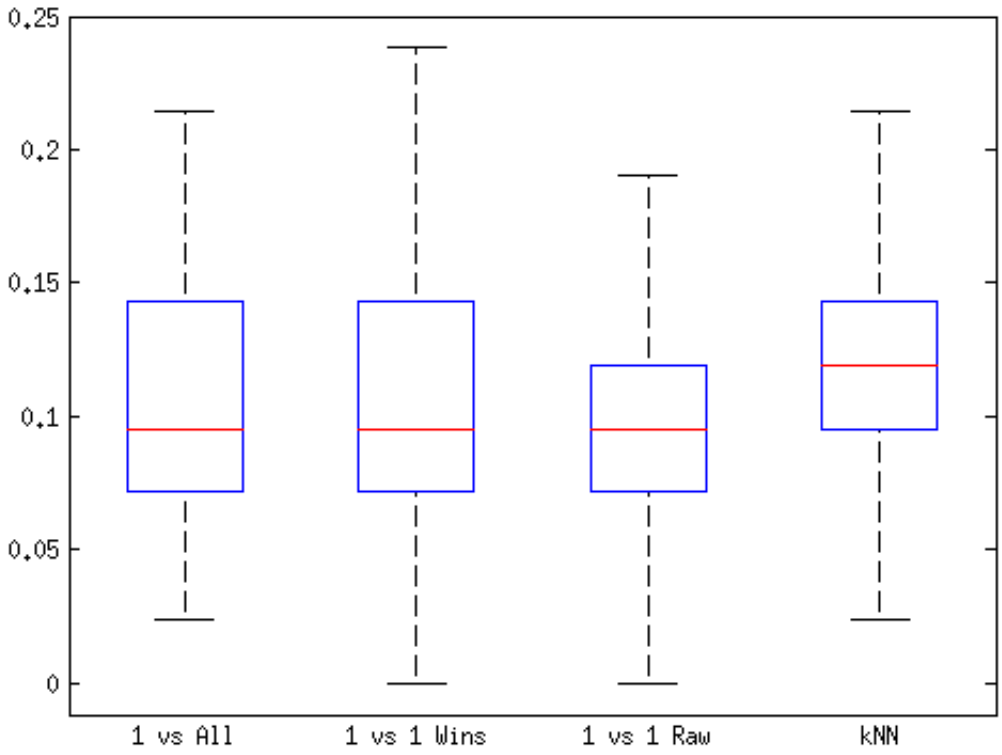

Figure 1: Box plots of the error rates on each classifier for the seeds data set

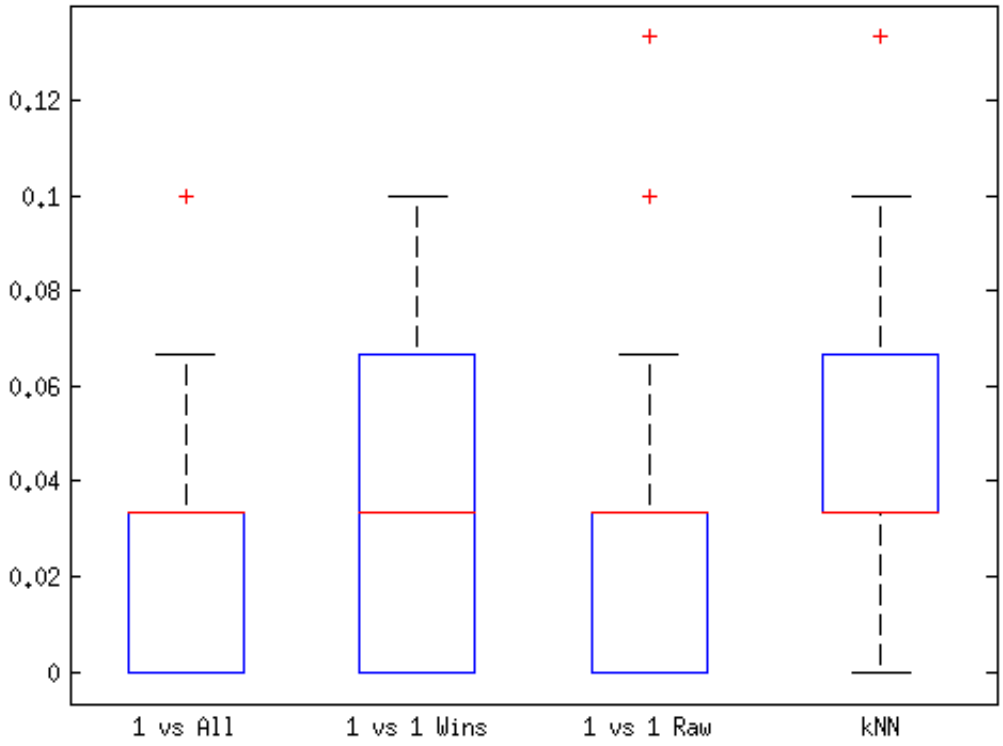

Figure 2: Box plots of the error rates on each classifier for the iris data set

performance. A further study should include data sets with more classes in order to evaluate which models perform better with higher class counts.

References

[1] Kai bo Duan and S. Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. In *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pages 278–285, 2005.

[2] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[3] Koby Crammer, Yoram Singer, Nello Cristianini, John Shawe-taylor, and Bob Williamson. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:2001, 2001.

[4] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[5] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[6] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling, 1998.

[7] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines, 2002.

[8] Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99:67–81, 2004.

[9] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[10] John C. Platt, Nello Cristianini, and John Shawe-taylor. Large margin dags for multiclass classification. In *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

[11] Hsuan tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on platt's probabilistic outputs for support vector machines, 2003.