Manipulation of Prior Probabilities in Support Vector Classification

Gavin C. Cawley and Nicola L. C. Talbot

School of Information Systems,
University of East Anglia,
Norwich, U.K.
gcc@sys.uea.ac.uk

Abstract

Asymmetric margin error costs for positive and negative examples are often cited as an efficient heuristic compensating for unrepresentative priors in training support vector classifiers. In this paper we show that this heuristic is well justified via simple resampling ideas applied to the dual Lagrangian defining the 1-norm soft-margin support vector machine. This observation also provides a simple expression for the asymptotically optimal ratio of margin error penalties, eliminating the need for the trial-and-error experimentation normally encountered. This method allows the use of a smaller, balanced training data set in problems characterised by widely disparate prior probabilities, reducing training time. We demonstrate the usefulness of this method on a real world benchmark problem, that of predicting forest cover type given only cartographic data.

1 Introduction

It is not uncommon in statistical pattern recognition tasks to encounter training data characterised by prior class probabilities that are not representative of the expected operational priors. This may be because the class priors vary in time or space, for instance the incidence of heart disease has been observed to vary geographically, due perhaps to differences in diet, or the frequency and severity of El Niño events may (or may not) be increasing with time due to climate change. Rather than include variables that capture the underlying causes of these variations, and risk incurring the curse of dimensionality, it may be more appropriate to modify the training procedure in some way to compensate for the difference in training set and operational priors in different situations.

Perhaps the most common reason for using a training set with unrepresentative priors is simply for computational expediency in classification problems with widely disparate prior probabilities, for instance the diagnosis of a rare medical disorder. In this case, a data set of a sufficient size to adequately characterise the statistical distribution of infrequently made positive diagnoses might require an enormous number of negative examples in order to accurately reflect the expected operational priors. Training time for many statistical classifiers can be greatly reduced in these circumstances if the classifier is trained on a smaller, balanced data set containing an equal number of positive and negative examples. The training algorithm must then be modified to compensate for the disparity in training set and operational priors.

Several techniques used to compensate for unrepresentative training set prior probabilities have been available for some time for conventional multi-layer feed-forward neural networks; approaches include scaling network outputs, resampling the training data, scaling weight updates and scaling target values used in training [1–3]. In this paper we describe a simple, computationally efficient modification to the Support Vector Machine (SVM) classifier, based on notional replication of training patterns, allowing adjustment of training set priors. This is shown to be equivalent to the well-known heuristic that assigns asymmetric margin error costs to positive and negative examples.

2 Support Vector Pattern Recognition

The support vector classification method [4, 5], given labelled training data,

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}, \quad \mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n, \quad y_i \in \{-1, +1\},$$

constructs a maximal margin linear classifier in a high dimensional feature space $\Phi(\mathbf{x})$, defined by a positive definite kernel function, $k(\mathbf{x}, \mathbf{x}')$, specifying an inner product in the feature space, $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$. The function implemented by a support vector machine is given by

$$f(\mathbf{x}) = \left\{ \sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \right\} - b. \qquad (1)$$

To find the optimal coefficients, α, of this expansion it is sufficient to maximise the functional,

$$W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j), \qquad (2)$$

in the non-negative quadrant,

$$0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, \ell,$$

subject to the constraint,

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0. \qquad (3)$$

C is a regularisation parameter, controlling a compromise between the complexity of the function implemented by the support vector machine and training set accuracy. Generally only a small number of Lagrange multipliers, α, will have non-zero values; the corresponding input patterns are known as support vectors. Let $\mathcal{I}$ be the index set of training patterns with non-bound Lagrange multipliers,

$$\mathcal{I} = \{ i \quad : \quad 0 < \alpha_i^0 < C \},$$

and similarly $\mathcal{J}$ represent the set of patterns with multipliers at the upper bound C,

$$\mathcal{J} = \{ i \quad : \quad \alpha_i^0 = C \}.$$

(1) can then be written as an expansion over support vectors,

$$f(\mathbf{x}) = \left\{ \sum_{i \in \{\mathcal{I}, \mathcal{J}\}} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \right\} - b.$$

For a full exposition of the support vector method, see Vapnik [6, 7], Christianini and Shawe-Taylor [8], or Burgess [9].

3 Dealing with Unrepresentative Priors

In this section we demonstrate a simple method to compensate for training data, $\mathcal{D}$, characterised by prior probabilities that are not representative of the expected operational priors. Consider a dataset $\mathcal{D}'$, consisting of ζ_i replicates of the i^{th} pattern comprising $\mathcal{D}$, $i = 1, 2, \ldots \ell$. Symmetry arguments indicate that two or more training patterns belonging to the same class and sharing the same input vector may also share Lagrange multipliers in the support vector expansion maximising the functional given in equation 2. Note there remains at most ℓ distinct Lagrange multipliers in the optimal support vector expansion for $\mathcal{D}'$. Therefore to find the optimal support vector expansion for this dataset, it is sufficient to maximise

$$W(\alpha) = \sum_{i=1}^{\ell} \zeta_i \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \zeta_i \zeta_j \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \qquad (4)$$

in the non-negative quadrant,

$$0 \leq \alpha_i \leq C, \qquad i = 1, 2, \ldots, \ell, \qquad (5)$$

subject to the modified constraint

$$\sum_{i=1}^{\ell} \zeta_i \alpha_i y_i = 0. \qquad (6)$$

The i^{th} support vector will be also be replicated ζ_i times, so the output of the support vector machine is as follows:

$$f(\mathbf{x}) = \left\{ \sum_{i=1}^{\ell} \zeta_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \right\} - b.$$

To compensate for a discrepancy between training set and operational priors, the weighting factor ζ_i, for the i^{th} training pattern of $\mathcal{D}$, is given by

$$\zeta_i = \frac{p_o(\mathcal{C}^i)}{p_t(\mathcal{C}^i)},$$

where $p_t(\mathcal{C}^i)$ is the *a-priori* probability of class $\mathcal{C}^i$ encountered in the training set and $p_o(\mathcal{C}^i)$ is the operational prior for $\mathcal{C}^i$, where $\mathcal{C}^i$ is the class to which the i^{th} training pattern belongs.

Note that following a change of coordinates, such that $\alpha_i^* = \alpha_i \zeta_i$, the solution to the optimisation problem given by equations 4-6 is equivalent to the original optimisation problem (equations 2 and 3, substituting α_i^* for all occurrences of α_i), subject to the the modified box constraint

$$0 \leq \alpha_i^* \leq \zeta_i C, \qquad i = 1, 2, \ldots, \ell.$$

The existing heuristic that assigns different margin error penalties to positive and negative training examples is therefore equivalent to resampling the training data to reflect the expected operational priors (c.f. [10, 11]).

3.1 Asymmetric Misclassification Costs

In the case of binary classification, for any risk functional that is a linear combination of penalties for each observation, the imposition of asymmetric false-positive and false-negative misclassification costs is equivalent to an unequal replication of positive and negative training examples. Consider a generalised empirical risk functional,

$$R^*_{\text{Emp}} = \frac{1}{\ell} \sum_{i=1}^{\ell} c_i \theta(y_i, f(\mathbf{x}_i, \alpha)), \tag{7}$$

where c_i is the cost associated with the error for pattern i. For binary pattern recognition, where $y_i, f \in \{-1, +1\}$, typically

$$\theta(y, f(\mathbf{x}, \alpha)) = \left\{ \begin{array}{ll} 0, & y = f(\mathbf{x}, \alpha) \\ 1, & y \neq f(\mathbf{x}, \alpha) \end{array} \right. .$$

To implement asymmetric misclassification costs for positive and negative examples,

$$c_i = \left\{ \begin{array}{ll} c^+ & y_i = +1 \\ c^- & y_i = -1 \end{array} \right. ,$$

where c^+ is the cost associated with false-negative and c^- the cost associated with false-positive misclassifications. Clearly the generalised risk functional given by equation 7 is equivalent to the standard empirical risk,

$$R_{\text{Emp}} = \frac{1}{\ell'} \sum_{i=1}^{\ell'} \theta(y_i, f(\mathbf{x}_i, \alpha)),$$

evaluated over a second dataset consisting of c^+ replicates of each positive training example and c^- replicates of each negative example.

4 Results

The Forest Cover Type benchmark problem, available from the UCI KDD Archive [12], provides a good test of methods used to compensate for unrepresentative training set priors. The classification task is to predict the forest cover type (seven classes), for 30m × 30m cells, given only cartographic data [13]. The input vector for each pattern is comprised of ten continuous variables (at least some of which are quantised) and two categorical variables; soil type (forty classes) and wilderness area designation (four types).

An interesting feature of this dataset is the great disparity in prior probabilities, as shown in table 1. The number of examples of the class best represented in the data set (Lodgepole Pine) is over two orders of magnitude greater than that of the least well represented (Cottonwood/Willow). As a result the originators of the dataset have partitioned the data into a training set, containing 1620 examples of each class, a validation set for model selection, containing 540 examples of each class, and a test set containing the remaining patterns. Support vector machines can be used to implement a near optimal classifier for this task, providing the disparity between training and test set priors can be accommodated. Note that due to the balanced selection of patterns for the training and validation sets, the prior probabilities encountered in the test set are even more unbalanced than the assumed population priors given in table 1. For example the prior probability for Lodgepole Pine increases to 0.4968, while the prior for Cottonwood/Willow falls to 0.001 (almost a five-fold reduction!). This gives some cause for concern that the test set incorporates a strong bias towards classes with a high prior probability.

Table 1: Number of examples and operational prior for each forest cover type.

Cover Type	Patterns	Prior
Spruce-Fir	211840	0.3646
Lodgepole Pine	283301	0.4876
Ponderosa Pine	35754	0.0615
Cottonwood/Willow	2747	0.0047
Aspen	9493	0.0163
Douglas-Fir	17367	0.0299
Krummholz	20510	0.0353
Total	**581012**	

As this is a multi-class problem, we adopt the pairwise classification approach [14, 15]. An n-class classifier is constructed from $k = n(n-1)/2$ two-class classifiers, one for each distinct pair of classes. An input pattern is classified as belonging to the class receiving the largest number of votes. The prior adjustment method can be applied to each two-class classifier on an individual basis. Before training, the continuous independent variables were standardised to have a zero mean and unit variance over the training set. The categorical variables remained in the raw 1-of-c coding scheme used in the benchmark dataset (a value of 1 indicating that a feature is present, a value of 0 indicating a feature is not present). A Gaussian radial basis function was used as the kernel function in each network,

$$k(\mathbf{x}, \mathbf{x}') = e^{-\gamma||\mathbf{x}-\mathbf{x}'||^2}.$$

Support vector machines trained using a range of values for both the kernel width, γ, and regularisation constant, C. The best networks for each of the 21 two-class problems were selected to minimise the number of classification errors over the validation set. For networks trained using prior adjustment, the errors on the validation set were also weighted according to the operational priors for consistency. All networks were trained using a MATLAB implementation [16] of a modified version of Platt's Sequential Minimal Optimisation (SMO) algorithm [17, 18].

The pairwise classifier constructed using the standard sequential minimal optimisation algorithm achieves an accuracy of 0.7220 over the test set. This result is a minor improvement over the figure of 0.71 quoted in Platt *et al.* [19]. This difference can be explained by the individual selection of kernel width and regularisation parameters for each two-class network. Table 2 shows a confusion matrix for the standard pairwise classifier over the test set, the Cottonwood/Willow class is of particular interest. A larger number of Ponderosa Pine patterns are misclassified as being Cottonwood/Willow than Cottonwood/Willow patterns are classified correctly. This is entirely due to unrepresentative training set priors. A similar story is true of Aspen/Lodgepole Pine.

The pairwise classifier constructed using the modified sequential minimal optimisation algorithm achieves an accuracy of 0.7341 over the test set. Table 3 shows a confusion matrix for this classifier compiled over the test set. Clearly the classes with the lowest prior probabilities now claim far fewer test set patterns belonging to more common classes, especially in the case of Cottonwood/Willow and Aspen, indicating the success of the prior adjustment procedure. It is important to note that the prior probabilities were adjusted in accordance with the assumed population priors given in table 1. There is however, an even greater disparity in prior class probabilities encountered in the test set, and so a greater improvement in test set performance might be expected if training set prior probabilities were adjusted to match the measured test set priors. This is indeed the case for Forest Cover dataset, achieving a test set accuracy of 0.7355. Of course the use of direct measurements from the test set during the training procedure is improper, as it corrupts the statistical purity of the test set.

This highlights a subtle error that can be introduced by selecting a balanced training set. We recommend that if a balanced training set is to be used, it should be selected as follows: first divide the available data into two segments. The first segment forms the test set; the second segment is used firstly to estimate the true operational priors, and the to draw the patterns used to form a balanced training set. This ensures that the test set remains representative of the true operational priors, while providing as estimate of the operational priors that is independent of the test set.

5 Summary

The main contribution of this paper is a simple justification of an existing heuristic used to compensate for unrepresentative training set prior class probabilities in support vector pattern recognition. It is shown that the use of unequal penalties for the margin errors associated with positive and negative examples is equivalent to a resampling of the training data such that it reflects the true operational priors. This observation also provides an expression for the optimal ratio of margin error penalties. It is also shown that the use of unequal misclassification costs [1, 20, 21] can be accommodated by resampling the training data and so can also be implemented by unequal margin error penalties. The effectiveness of this procedure is demonstrated on a large, real world pattern classification task, the UCI cover type prediction problem. The test set accuracy of 0.7341 appears to be the best result achieved for this benchmark to date.

If the proposed method is used to permit faster training using a smaller, balanced training set, care must be taken in selecting the examples drawn from each class. For support vector machines we would like to select those patterns closest to the optimal decision surfaces between each class. We are faced with a trade-off between achieving adequate coverage of the class boundaries (improving generalisation) and minimising size of the training set (improving training time).

The technique we have described assumes that the empirical distribution implied by the training data is an accurate representation of the underlying distribution from which the data were drawn. Clearly the precision with which *a-priori* probabilities are adjusted might be expected to improve asymptotically as the size of the training set increases.

Acknowledgements. The authors would like to thank Rob Foxall and Danilo Mandic for their helpful comments on previous drafts of this paper and Steve Dorling for advice regarding El Niño.

References

[1] D. Lowe and A. R. Webb. Exploiting prior knowledge in network optimization: an illustration from medical prognosis. *Network: Computation in Neural Systems*, 1(3):299–323, 1990.

Table 2: Confusion matrix for the standard pairwise SVM classifier for the UCI forest cover type benchmark over the test partition

	Spruce	Lodgepole	Ponderosa	Cottonwood	Aspen	Douglas	Krummholz	
Actual	151012	43086	202	0	2334	573	12473	**Spruce**
	58232	194655	5293	96	14013	6906	1946	**Lodgepole**
	14	696	26539	1004	400	4941	0	**Ponderosa**
	0	0	41	530	0	16	0	**Cottonwood**
	127	477	96	0	6585	47	1	**Aspen**
	31	376	2554	321	152	11773	0	**Douglas**
	776	95	0	0	1	0	17478	**Krummholz**

Predicted

Table 3: Confusion matrix for the prior-adjusted pairwise SVM classifier for the UCI forest cover type benchmark over the test partition

	Spruce	Lodgepole	Ponderosa	Cottonwood	Aspen	Douglas	Krummholz	
Actual	151946	42831	302	0	1063	337	13201	**Spruce**
	60032	202093	6698	4	5998	4261	2055	**Lodgepole**
	31	763	30608	208	34	1950	0	**Ponderosa**
	0	2	232	313	0	40	0	**Cottonwood**
	437	1789	326	0	4571	196	14	**Aspen**
	55	957	5627	73	45	8450	0	**Douglas**
	805	110	0	0	0	0	17435	**Krummholz**

Predicted

[2] D. W. McMichael. Structural generalization in neural classification: Incorporation of prior probabilities. In *Proceedings of the I.E.E. Colloquium on Adaptive Filtering, Non-Linear Dynamics and Neural Networks*, pages 6/1–6/5, 1991.

[3] S. Lawrence, I. Burns, A. Back, Ah Chung Tsoi, and C. Lee Giles. Neural network classification and prior class probabilities. In G. Orr, K.-R. Müller, and R. Caruana, editors, *Tricks of the Trade, Lecture Notes in Computer Science State-of-the-Art Surveys*, pages 299–314. Springer-Verlag, 1998.

[4] B. Boser, I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory*, pages 144–152, Pittsburgh, 1992. ACM.

[5] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[6] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[7] V. N. Vapnik. *Statistical Learning Theory.* Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications and Control. Wiley, New York, 1998.

[8] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods).* Cambridge University Press, Cambridge, U.K., 2000.

[9] C. J. C. Burgess. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167, 1998.

[10] Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. Technical Report 1016, Department of Statistics, University of Wisconsin, 1210 West Drayton St., Madison, WI 53706, March 2000.

[11] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on Artificial Intelligence (Workshop ML3)*, pages 17–21, Stockholm, Sweeden, 1999.

[12] S. D. Bay. The UCI KDD archive [`http://kdd.ics.uci.edu/`]. University of California, Department of Information and Computer Science, Irvine, CA, 1999.

[13] J. A. Blackard and D. J. Dean. Comparative accuracies of artificial neural networks and discriminative analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24:131–151, 1999.

[14] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: A stepwise procedure for building and training a neural network. In F. Fogelman-Soulie and J. Herault, editors, *Neurocomputing: Algorithms, Architectures and Applications,* NATO ASI. Springer, 1990.

[15] U. Kreßel. Pairwise classification and support vector machines. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 15, pages 255–268. MIT Press, Cambridge, Massachusetts, 1999.

[16] G. C. Cawley. MATLAB SVM toolbox [`http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox`]. University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000.

[17] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[18] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 12, pages 185–208. MIT Press, Cambridge, Massachusetts, 1999.

[19] J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 547–553. MIT Press, 2000.

[20] M. D. Richard and R. P. Lippmann. Neural network classifiers estimate Bayesian a-posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.

[21] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.