On the Explicit Use of Example Weights in the Construction of Classifiers

Andrew Naish-Guzman, Sean Holden, and Ulrich Paquet

Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, UK
andrew.naish-guzman, sean.holden, ulrich.paquet@cl.cam.ac.uk

Abstract. We present a novel approach to two-class classification, in which a classifier is parameterised in terms of a distribution over examples. The optimal distribution is determined by the solution of a linear program; it is found experimentally to be highly sparse, and to yield a classifier resistant to noise, whose error rates are competitive with the best existing methods.

1 Introduction

Many classification algorithms associate a weight with each element of the training set. In support vector machines, these weights are Lagrange multipliers in a quadratic optimisation problem; when set correctly, they define a separating hyperplane in the kernel-induced feature space (Schölkopf et al. (1999)). The relevance vector machine (Tipping (2001)) places a Gaussian of constant width over every data point and, in a Bayesian setting, assigns a weight to each such basis function. By an explicit assumption on the form of the solution, the distribution of weights is encouraged to be sparse. Boosting methods, in contrast, work iteratively and update the weights in response to each hypothesis chosen by a so-called *weak learner* (Freund and Schapire (1995)). An example's weight is related to the frequency with which it has been misclassified; by appropriate reweighting of the data, boosting algorithms encourage the weak learner to explore advantageous regions of hypothesis space.

While studying the behaviour of boosting when applied to a simple weak learner, we observed the approximate convergence of the example weights, and the correlated convergence of the decision boundary. This observation motivated the idea that a *fixed* distribution over examples may be capable of inducing a useful distribution over the basis class. In this work, we show how a novel interpretation of example weights may indeed yield a sensible distribution over hypotheses. The optimal weight assignment is given by the solution of a linear program, and we show that the predictions of this distributed classifier may then be evaluated efficiently. Preliminary results indicate our algorithm is stable in noisy conditions, and performs competitively with the best existing methods. It also yields sparse solutions, in that many examples are given weights equal to or very close to zero. The relevance vector machine also exhibits this property, but is computationally more involved, and shows considerable sensitivity to the parameter that governs the width of each Gaussian.

2 Interpretation of weights

Let us formalise the problem. We have a data set $D = \{(x_i, y_i)\}_{i=1}^{m}$, where $x_i \in X$ and $y_i \in Y = \{\pm 1\}$, a distribution over examples d_i, such that $\|\mathbf{d}\|_1 = 1$ and $d_i \geq 0$, and also a class of hypotheses $\mathcal{H}$ with an associated measure, allowing us to place over it a distribution $p(h)$. We will find that even for uniform $p(h)$, a novel interpretation of $\mathbf{d}$ has the potential to yield a complex distribution over the basis class.

Consider the following scheme for classifying a new point $x \in X$. We draw examples from D according to the probability vector $\mathbf{d}$; for each example (x_i, y_i) selected, we sample a hypothesis from $\mathcal{H}$ according to $p(h)$, with the restriction that $h(x_i) = y_i$, and evaluate $h(x)$. If we sum indefinitely many such classifications and normalize the result, the final output will tend to

$$\begin{aligned} F_{\mathbf{d}}(x) &= \sum_{i=1}^{m} d_i \int \mathbb{I}[h(x_i) = y_i] h(x) dp(h) \\ &= \mathbb{E}_{(x_i, y_i) \sim \mathbf{d}} \left[\mathbb{E}_{h : h(x_i) = y \cdot i} [h(x)] \right] . \qquad (1) \end{aligned}$$

Assume that $p(h)$ is symmetric with respect to the two classes; that is, we have $p(h(x) = 1) = p(h(x) = -1)$ for all x. We may now classify x:

$$\begin{aligned} F_{\mathbf{d}}(x) &= \sum_{i=1}^{m} d_i \int \mathbb{I}[h(x_i) = y_i] h(x) dp(h) \\ &= \sum_{i=1}^{m} d_i \left(\int \mathbb{I}[h(x) = h(x_i) = y_i] y_i dp(h) - \int \mathbb{I}[h(x) \neq h(x_i) = y_i] y_i dp(h) \right) \\ &= \sum_{i=1}^{m} d_i y_i \left(\int \mathbb{I}[h(x_i) = y_i] dp(h) - 2 \int \mathbb{I}[h(x) \neq h(x_i) = y_i] dp(h) \right) \\ &= \sum_{i=1}^{m} d_i y_i \left(\frac{1}{2} - \int \mathbb{I}[h(x) \neq h(x_i)] dp(h) \right) , \qquad (2) \end{aligned}$$

where in the last line we have used the symmetry of $p(h)$. We note that in (2), the final bracketed expression has the appearance of a kernel function: it is related to the probability that an arbitrary hypothesis drawn from $p(h)$ will have equal sign evaluated at x and x_i.

2.1 Assignment of weights

Write the *margin* of the classifier on each element of the training set as a vector:

$$\begin{aligned} [y_j F_{\mathbf{d}}(x_j)]_{j=1}^{m} &= \left[y_j \sum_{i=1}^{m} d_i y_i \left(\frac{1}{2} - \int \mathbb{I}[h(x_j) \neq h(x_i)] p(h) dh \right) \right]_{j=1}^{m} \\ &= \mathbf{d}^{\top} \mathsf{Q}, \qquad (3) \end{aligned}$$

where

$$\mathsf{Q}_{ij} = y_i y_j \left(\frac{1}{2} - \int \mathbb{I}[h(x_j) \neq h(x_i)] p(h) dh \right) .$$

Q is symmetric; we have also $\mathsf{Q}_{ii} = \frac{1}{2}$ for all i. The linear formulation (3) allows us to find suitable weights by solving a linear program. For example, we can choose weights that maximise the minimum margin over the training set:

$$\begin{aligned} &\max_{\mathbf{d},\gamma} \gamma \\ \text{subject to } &y_i F_{\mathbf{d}}(x_i) \geq \gamma \text{ for } i = 1 \ldots m \\ &d_i \geq 0 \text{ and } \|\mathbf{d}\|_1 = 1. \end{aligned} \tag{4}$$

Alternatively, we can introduce a parameter $C > 0$ and slack variables $\boldsymbol{\xi}$ to allow a small number of misclassifications:

$$\begin{aligned} &\max_{\mathbf{d},\boldsymbol{\xi},\gamma} \gamma - C \sum_{i=1}^{m} \xi_i \\ \text{subject to } &y_i F_{\mathbf{d}}(x_i) \geq \gamma - \xi_i \text{ for } i = 1 \ldots m \\ &d_i \geq 0 \text{ and } \|\mathbf{d}\|_1 = 1 \\ &\xi_i \geq 0. \end{aligned} \tag{5}$$

In either case, the final classifier is given by $\text{sgn}(F_{\mathbf{d}}(x))$.

Optimisation with respect to a weight vector's 1-norm was investigated in the context of support vector machines (for which the 2-norm is more conventional) by Bradley and Mangasarian (1998), and Zhu et al. (2003). The approach we have adopted is differentiated by our probabilistic interpretation of $\mathbf{d}$, which yields a finite set of bases (2) that correspond implicitly to an integral over $\mathcal{H}$.

3 Implementation

To construct the matrix Q, we need to choose $\mathcal{H}$ and $p(h)$, and thus determine

$$\int \mathbb{I}[h(x_i) \neq h(x_j)] dp(h) \qquad \text{for } i, j = 1, 2, \ldots, m. \tag{6}$$

In the following analysis, we restrict our attention to the two-dimensional case, and fix $\mathcal{H}$ to be the class of planes in $\mathbb{R}^2$. Extending these concepts to higher dimensions and further classes is deferred to future work.

Without loss of generality, let the mean of the data be at the origin $O = (0, 0)$, and let all training coordinates lie in the region $[-R, R]^2$. All hypotheses $h \in \mathcal{H}$, with the exception of those that pass through the origin, may be paramaterised by a pair $(\mathbf{r}, s) \in (\mathbb{R}^2, \{\pm 1\})$. The coordinate $\mathbf{r}$ indicates the closest point on the line to O, while the sign term s defines the classification of the origin. Let us now define a measure on $\mathcal{H}$ by placing a uniform distribution over $\mathbf{r}$ in the range $[-R, R]^2$, and assigning equiprobably $s = 1$ or $s = -1$.

In order to calculate (6), we must find the expected proportion of hypotheses discriminating between x_i and x_j. With the preceding assumptions, we may now consider (6) as the volume of parameter space $\mathcal{H}' \subseteq \mathcal{H}$, in which $h(x_i) \neq h(x_j) \Leftrightarrow h \in \mathcal{H}'$. The situation is illustrated in Figure 1. We note that, for a given pair $(\mathbf{r}, s)$, if the hypothesis parameterised by $(\mathbf{r}, s)$ satisfies this property, so also will that parameterised by $(\mathbf{r}, -s)$.

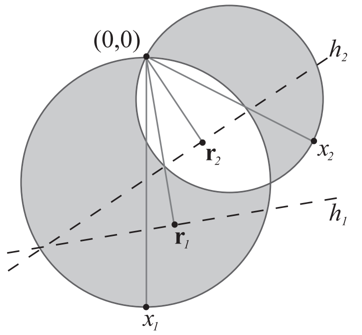

Fig. 1. Visualisation of (6). The shaded region parameterises hypotheses $h \in \mathcal{H}' \subseteq \mathcal{H}$ for which $h(x_1) \neq h(x_2) \Leftrightarrow h \in \mathcal{H}'$. Two hypotheses are shown, h_1 and h_2, parameterised by $\mathbf{r}_1$ and $\mathbf{r}_2$ respectively. Independently of s, $h_1 \in \mathcal{H}'$ discriminates between x_1 and x_2, while $h_2 \notin \mathcal{H}'$ classifies the two examples identically.

For a point $x \in X$, write the circular region parameterising hypotheses that discriminate between x and O as $\bigcirc_x$. Now,

$$\int \mathbb{I}[h(x_i) \neq h(x_j)]dp(h) \propto \left|\bigcirc_{x_i} \setminus \bigcirc_{x_j}\right| + \left|\bigcirc_{x_j} \setminus \bigcirc_{x_i}\right|$$
$$= \left|\bigcirc_{x_i}\right| + \left|\bigcirc_{x_j}\right| - 2\left|\bigcirc_{x_i} \cap \bigcirc_{x_j}\right| .$$

It can be shown that the area of intersection $\left|\bigcirc_{x_i} \cap \bigcirc_{x_j}\right|$ is given by

$$\frac{1}{2}\Big(\|x_i\|^2(\theta_i - \sin\theta_i) + \|x_j\|^2(\theta_j - \sin\theta_j)\Big),$$

where θ_i (θ_j) is the angle subtended at the centre of $\bigcirc_{x_i}$ ($\bigcirc_{x_j}$) by radii extending to the two points of intersection. Define

$$A(x_i, x_j) = \|x_i\|^2\left(\pi - \theta_i + \sin\theta_i\right) + \|x_j\|^2\left(\pi - \theta_j + \sin\theta_j\right),$$

so that

$$\mathsf{Q}_{ij} = y_i y_j \left(\frac{1}{2} - \frac{1}{4R^2}A(x_i, x_j)\right). \tag{7}$$

4 Results

Results were obtained for two benchmark data sets: Ripley's mixture of Gaussians (Ripley (1996)) consists of 250 training examples and 1000 test examples; the Banana set[1] consists of 100 realisations of 400 training examples and 4900

[1] Available from `http://ida.first.fhg.de/projects/bench/benchmarks.htm`.

test examples. Both are two-dimensional. We chose the parameter C on the Ripley set by examining the decision boundary for a variety of choices, and selecting the one with qualitatively best fit; this was found to be $C = 0.009$. For the Banana benchmark, we split the training set into equal subsets for training and validation, to find the optimal $C \in \{0.01, 0.012, \ldots, 0.02\}$. In each case, we used the formulation (7), and set $R = 5$.

On Ripley's set, the test error was 8.6%. This compares favourably with existing methods: using an SVM, Ripley achieved 10.6%, while Tipping's RVM achieved 9.3%.[2] The Bayes rate is around 8%. Over the first ten realisations of the Banana set, our method achieved a mean test error of 10.9%; the support and relevance vector machines obtained error rates of 10.6% and 10.5% respectively.[3]

The decision boundary we obtained on the Ripley set is illustrated in Figure 2. The training data are shown, together with surrounding circles, each of whose radii is proportional to the weight of the associated data point. It is interesting to observe that many components of this distribution are equal to or close to zero, and that the heavily weighted examples tend to be some distance from the decision boundary. The SVM solution to this problem used 38 support vectors, while the RVM solution used 4 relevance vectors; our solution places non-zero weight on 8 examples.

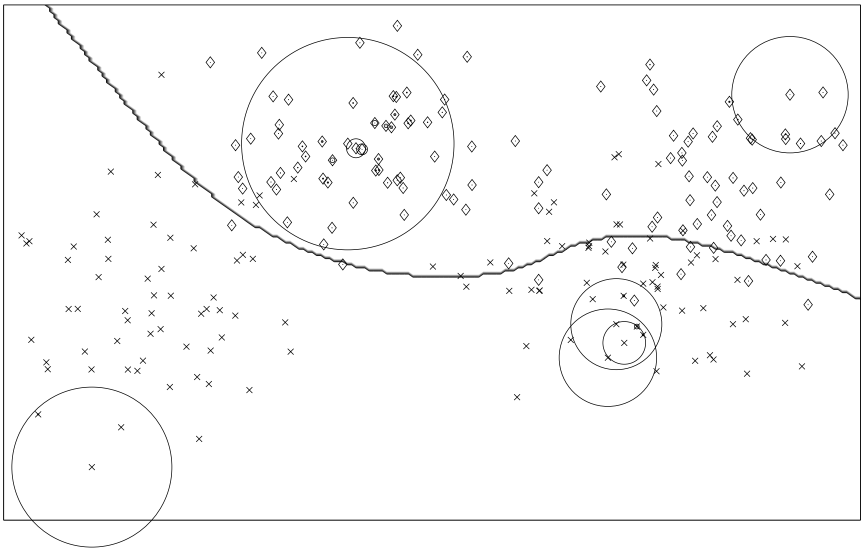

Fig. 2. The decision boundary obtained on Ripley's 2-d training set by solving (5) with $C = 0.009$. Data points with non-zero weight assignment have been circled; the radius of the circle is proportional to the example's weight.

[2] Results from Bishop and Tipping (2003).

[3] Results from Tipping (2001).

5 Conclusions

We have shown how a simple sampling scheme for classification and a novel interpretation of weighted examples induces a distribution over hypothesis space. We have evaluated the predictions of this distributed classifier for an optimal weighting of the training set, and found these predictions to be resistant to overfitting. Our method has certain advantages. The weight assignment can be determined easily by solving a linear program, with a single parameter defining the degree to which misclassifications are tolerated. The weight vector is experimentally found to be sparse when the solution has not overfit; new classifications are then possible in time $\mathcal{O}(m')$, where $m' \leq m$ is the number of examples in the training set with non-zero weight. We have observed also that our "support vectors" lie away from the decision boundary and tend to be fewer in number than for an SVM solution. Unfortunately, we have not provided a rigorous explanation for our algorithm's strong performance. We believe its success is due to the calculation of a Bayesian integral under a novel noise model; developing the theory to support this hypothesis, and extending the algorithm to further dimensions, are areas of current research.

6 Acknowledgments

Andrew Naish-Guzman is supported by a Millennium Scholarship. Ulrich Paquet is supported by a Commonwealth Scholarship, and expresses his thanks to the Commonwealth Scholarship Commission.

References

Bishop, C. M and Tipping, M. E. (2003) Bayesian regression and classification. In J. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, *Advances in Learning Theory: Methods, Models and Applications 190.* IOS Press, Amsterdam.

Bradley, P. and Mangasarian, O. (1998). Feature selection via concave minimization and support vector machines. In J. Schavlik, *International Conference on Machine Learning '98.* Morgan Kaufmann.

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory*, 23–37.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press.

Schölkopf, B., C. J. C. Burges, and A. J. Smola (1999). *Advances in Kernel Methods: Support Vector Learning.* MIT Press.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research 1*, 211–244.

Zhu, J., Rosset, S., Hastie, T. and Tibshirani, R. (2003) 1-norm support vector machines. *Neural Information Processing Systems 16.*