A SPARSE BAYESIAN COMPRESSION SCHEME — THE INFORMATIVE VECTOR MACHINE

NEIL D. LAWRENCE AND RALF HERBRICH

1. INTRODUCTION

Kernel based learning algorithms allow the mapping of data-set into an infinite dimensional feature space in which a classification may be performed. As such kernel methods represent a powerful approach to the solution of many non-linear problems. However kernel methods do suffer from one unfortunate drawback, the Gram matrix contains m rows and columns where m is the number of data-points. Many operations are precluded (e.g. matrix inverse $O(m^3)$) when data-sets containing more than about 10^4 points are encountered. One approach to resolving these issues is to look for sparse representations of the data-set [7, 5, 2]. A sparse representation contains a reduced number of examples.

Loosely speaking we are interested in extracting the maximum amount of information from the minimum number of data-points. To achieve this in a principled manner we are interested in estimating the amount of information each data-point contains. In the framework presented here we make use of the Bayesian methodology to determine how much information is gained from each data-point.

2. BAYESIAN ON-LINE LEARNING AND COMPRESSION SCHEMES

In Bayesian learning we start with a prior, $p(\mathbf{w})$, over our parameter space. Learning $\mathbf{w}$ then proceeds by utilising Bayes's rule to obtain a posterior distribution over the parameter space, $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$, where $\mathbf{X}$ is a matrix of input data and $\mathbf{y}$ is a vector of class labels:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})\, p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})}.$$

A classical measure of the information a distribution $p(\cdot)$ contains is its negative entropy [1], $\mathcal{I}(p(\cdot)) = \int p(\mathbf{w}) \ln p(\mathbf{w})\, d\mathbf{w}$. Our new knowledge about the parameters is represented by the posterior distribution and its information content, $\mathcal{I}(p(\mathbf{w}|\mathbf{y}, \mathbf{X}))$. If our data-set consists of training examples $(\mathbf{x}_i, y_i) \in (\mathcal{F} \times \{-1, +1\})$ and we may decompose the inference process,

$$p(\mathbf{w}|\mathbf{X}_i, \mathbf{y}_i) = \frac{p(y_i|\mathbf{x}_i, \mathbf{w})\, p(\mathbf{w}|\mathbf{X}_{i-1}, \mathbf{y}_{i-1})}{p(y_i|\mathbf{x}_i)} \tag{1}$$

where $p(\mathbf{w}|\mathbf{y}, \mathbf{X}) \equiv p(\mathbf{w}|\mathbf{X}_m, \mathbf{y}_m)$, $\mathbf{X}_i$ is a matrix containing the first i data-points and $\mathbf{y}_i$ is a vector containing the first i labels. The use of this decomposition is common in the context of on-line learning [2] where the posterior is updated as each data-point arrives. The advantage of utilising the above decomposition is that we evaluate the information gain associated with the incorporation of every data-point.

In this work we study sparse representations in the context of linear classification models. Hence, the predicted output of the model — the likelihood model — is taken to be

$p(y_i|\mathbf{x}_i, \mathbf{w}) = g\left(y_i \mathbf{x}^{\mathrm{T}} \mathbf{w}\right)$ where

$$g(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt\,.$$

If we take our prior over the parameters $\mathbf{w}$ to be Gaussian, $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, \mathbf{\Sigma})$, our posterior distribution after observing one data-point is analytically tractable has the form of a 'squashed Gaussian'. However, observing a further data-point will lead us to an intractable form for the new posterior distribution. To finesse this problem we follow [2] in approximating the posterior at every iteration i with a Gaussian distribution $p^{(i)}(\mathbf{w})$ through minimisation of the Kullback-Leibler distance between the two distributions.

The important difference of our work to [2] is in the selection of the next data-point $(\mathbf{x}_k, y_k)$ to be considered for an update of the posterior distribution $p^{(i-1)}(\mathbf{w})$ over weight space. For every remaining data-point $(\mathbf{x}_j, y_j)$ we compute the Gaussian approximation $\widehat{p}^{(j)}(\mathbf{w})$ to the 'squashed Gaussian' posterior distribution and determine the *informative value* $\mathcal{I}\left(\widehat{p}^{(j)}(\cdot)\right)$. Then, the data-point $(\mathbf{x}_k, y_k)$ with the highest informative value is selected and the distribution $p^{(i-1)}(\mathbf{w})$ is updated to $p^{(i)}(\mathbf{w}) = \widehat{p}^{(k)}(\mathbf{w})$.

In order to achieve a high sparsity (a small number of updates) together with a high generalisation performance at the same time we appeal to PAC generalisation error bounds known for compression schemes. In a nutshell, an algorithm is called a *compression scheme* if the final classifier can be reconstructed from only a small subset of the training sample[1]. If it happens to be the case that the resulting classifier also has a small training error then we can guarantee a small generalisation error without any further assumptions on the data distribution. The main result reads as follows.

Theorem 1. *For all distributions $p((\mathbf{x}, y))$ over the input-output space, for all $\delta \in (0, 1]$ with probability at least $1 - \delta$ over the random draw of the training sample of size m, if a compression scheme only uses a subset of the training sample of size d then the misclassification probability of the resulting classifier cannot exceed*

$$\frac{m}{m-d}\hat{R} + \sqrt{\frac{1}{2(m-d)}\left(d\ln\left(\frac{em}{d}\right) + 2\ln(m) + \ln\left(\frac{1}{\delta}\right)\right)} \tag{2}$$

where $\hat{R}$ is the misclassification rate on the training sample.

The application of (2) in our algorithm is either

(1) as a stopping criterion, that is we do not continue updating the posterior over weight space as soon as (2) starts increasing, or
(2) as a selection criterion among all the m different posterior distributions $p^{(i)}(\mathbf{w})$. Here, the training error $\hat{R}$ incurred by $p^{(i)}(\mathbf{w})$ is traded-off against $d = i$ in (2). This method is computationally more expensive but has the advantage of finding the sparsest solution which has still a good generalisation performance.

It is important to observe that for both methods the resulting algorithm is a compression scheme because the deletion of the less informative examples does not change the order at which the most informative training examples occur.

[1] For example, the relevance vector machine [6] is *not* a compression scheme as it requires the inner products to *all* the training examples for the estimation of a small number of non-zero expansion coefficients of the weight vector

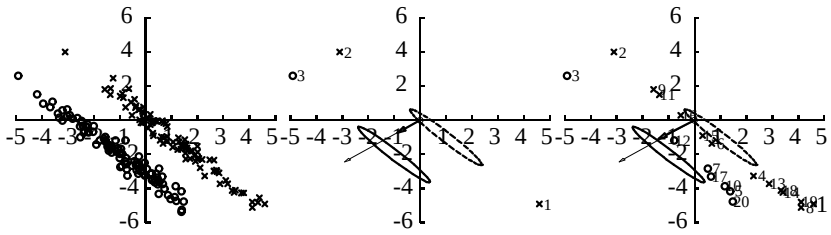

FIGURE 1. Results for the separable data. The Gaussian distributions which generated the points are marked as ellipses which mark one standard deviation from the centre. Class $y = 1$ is marked with a solid line and data-points from that distribution are marked as circles. Class $y = -1$ is marked as a dashed line with data-points from that distribution marked as crosses. The vector $\mathbf{w}$ learned by the algorithm is shown as a thick arrow. The optimum vector is shown as a thin arrow. The plot on the *left* shows the generated data-set. The *middle* plot shows the 3 'most informative' data-points and the order in which they were selected. The *rightmost* plot shows the 20 'most informative' data-points.

TABLE 1. Summary of results from the two experiments. The table shows the lowest value of the compression bound, the associated test error and the Bayes error. The numbers in brackets indicate, where appropriate, the associated number of data-points utilised for training.

	LOWEST BOUND	GENERALISATION ERROR	BAYES ERROR
SEPARABLE	0.272195 $(d = 2)$	7.3660×10^{-4}	2.8208×10^{-6}
OVERLAPPING	0.442134 $(d = 9)$	0.093625	0.0786

3. EXPERIMENTAL RESULTS

Consider a two class classification problem, where the class conditional densities are taken to be Gaussians with means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_{-1}$ and a *shared* covariance, $\mathbf{C}$. For a problem of this sort the Bayes optimal decision boundary is linear and may be computed in a straightforward manner. We considered two classification tasks for illustrating our approach.

Separable Classes First we considered a classification problem, depicted in Figure 1, where the two classes were well separated. A data-set was generated containing $m = 200$ training points. The compression bound was evaluated for all different values of d and was found to be at a minimum for $d = 3$. Table 3 shows various error rates of interest including the true generalisation error of the selected classifier and the Bayes optimal error rate, both of which may be computed as the true class conditional densities are known. The minimum achieved generalisation error was observed to be 2.9229×10^{-5}. It is interesting to note that this was achieved by the classifier based on the 11 most informative data-points, i.e. the best generalisation error was achieved using only 5.5 % of the data-points.

Overlapping Classes Results for a toy problem where the class conditional denisites overlap are shown in 2. Again various error rates are shown in 3. For this data set, the minimum generalisation error, 0.0793, was achieved with the 13 most informative data-points.

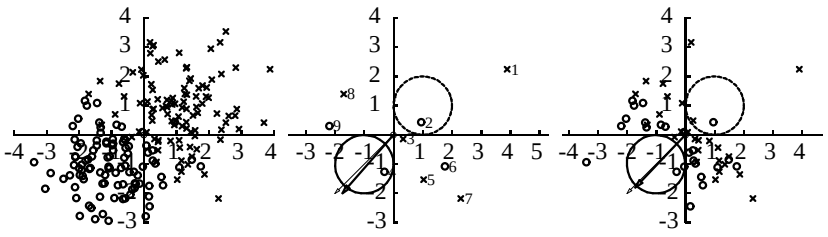

FIGURE 2. Plots for the toy problem with over-lapping classes. The *left-most* plot shows the 200 data-points, the *middle* plot shows the 9 'most informative' data-points and the order in which they were selected. The plot on the *right* shows the fifty 'most informative' data-points without ordering.

4. DISCUSSION

In this paper we have presented a Bayesian approach to compression schemes. Using the entropy of the Gaussian approximated posterior distribution as a selection criterion has shown to perform very well in preliminary experiments. The usage of compression bounds as a stopping criterion guarantees generalisation performance while finding a sparse approximation of the Bayes classifier. It is worth mentioning that the idea of entropy-based selection of training examples has been mentioned by other authors (see, e.g. [4]), but to our best knowledge, not for the case of classification with linear functions.

Future research is focused on casting our selection criterion into a Gaussian process framework [2] which would enable us to explicitly use kernel functions. For illustrative purposes the examples we have provided here are for the linear case; we expect to provide a kernelization at the workshop.

REFERENCES

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
[2] L. Csató and M. Opper. Sparse representation for gaussian process models. In Leen et al. [3].
[3] T. K. Leen, T. G. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA, 2001. MIT Press.
[4] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.
[5] A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In P. Langley, editor, *Proceedings of the International Conference in Machine Learning*, volume 17, pages 911–918, San Francisco, CA, 2000. Morgan Kauffman.
[6] M. E. Tipping. The relevance vector machine. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 652–658, Cambridge, MA, 2000. MIT Press.
[7] C. K. I. Williams and M. Seeger. Using the nystrom method to speed up kernel machines. In Leen et al. [3].