

---

# PAC-Bayes Analysis Of Maximum Entropy Learning

---

**John Shawe-Taylor and David R. Hardoon**

Centre for Computational Statistics and Machine Learning

Department of Computer Science

University College London, UK, WC1E 6BT

{j.shawe-taylor,d.hardoon}@cs.ucl.ac.uk

## Abstract

We extend and apply the PAC-Bayes theorem to the analysis of maximum entropy learning by considering maximum entropy classification. The theory introduces a multiple sampling technique that controls an effective margin of the bound. We further develop a dual implementation of the convex optimisation that optimises the bound. This algorithm is tested on some simple datasets and the value of the bound compared with the test error.

## 1 INTRODUCTION

Maximising the entropy of a distribution subject to certain constraints is a well-established method of regularising learning or general modelling both in statistics (Kapur & Keshavan, 1992) and machine learning (Wang et al., 2004; Dudik & Schapire, 2006). The motivation for the approach is that maximising the entropy is a method of preventing overspecialisation and hence overfitting of the model. Despite this clear motivation for and interest in the technique there are to our knowledge no statistical learning theory results that bound the performance of such heuristics and hence motivate the use of this approach.

There is an intriguing suggestion that the KL divergence appearing in the PAC-Bayes bound (Langford, 2005) could relate to the entropy of the distribution. It was this possibility that motivated the work presented in this paper. The difficulty with applying the PAC-Bayes framework was that there did not seem to be a natural way to define the distribution over hypotheses in such a way that the KL divergence measured

the entropy of the distribution, while the error would relate to the loss of the corresponding function.

We make this connection through a novel method of using the distribution to generate outputs that on the one hand ensures the probability of misclassification can be bounded by twice the stochastic loss, while on the other hand the empirical loss is related to the margin on the training data. Note that we concentrate on the classification case, though we believe that this work will lay the foundations for the more interesting problem of density function modelling using the maximum entropy principle (Dudik & Schapire, 2006).

While this work does not prove that using maximum entropy regularisation is preferable to other methods currently implemented in commonly used algorithms, as for example 2-norm regularisation in Support Vector Machines, or 1-norm regularisation in boosting and LASSO methods, it places this principle on a firm foundation in statistical learning theory and in this sense places it on a par with these other methods. We do present very preliminary experiments, but the question of whether maximum entropy regularisation as an approach to classification has a significant role to play in practical machine learning is left for future research.

The paper is organised as follows. Section 2 introduces the PAC-Bayes approach to bounding generalisation and gives the application to bounding error in terms of the entropy of the posterior distribution. Section 3 takes the results of this analysis to create a convex optimisation for which a dual form is derived that can be implemented efficiently. In Section 4 we present results comparing the bound values with test set errors for some UCI datasets. Despite the inclusion of these results we wish to emphasise that the main contribution of the paper is the development of new techniques that enable the PAC-Bayes analysis to be used to analyse a very different learning algorithm and promises to enable its application to more general modelling tasks.

## 2 ERROR ANALYSIS

We first state the general PAC-Bayes result following (McAllester, 1998, 1999; Seeger, 2003; Langford, 2005) after giving two relevant definitions. We assume a class  $C$  of classifiers with a prior distribution  $P$  over  $C$  and posterior distribution  $Q$  and a distribution  $\mathcal{D}$  governing the generation of the input/output samples. We use  $e_Q$  to denote the expected error under the distribution  $Q$  over classifiers:

$$e_Q = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}, c \sim Q} [I [c(\mathbf{x}) \neq y]].$$

Given a training sample  $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ , we similarly define

$$\hat{e}_Q = \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{c \sim Q} [I [c(\mathbf{x}_i) \neq y_i]].$$

**Theorem 2.1 ((Langford, 2005))** *Fix an arbitrary  $\mathcal{D}$ , arbitrary prior  $P$ , and confidence  $\delta$ , then with probability at least  $1 - \delta$  over samples  $S \sim \mathcal{D}^m$ , all posteriors  $Q$  satisfy*

$$\text{KL}(\hat{e}_Q \| e_Q) \leq \frac{\text{KL}(Q \| P) + \ln((m+1)/\delta)}{m}$$

where KL is the KL divergence between distributions

$$\text{KL}(Q \| P) = \mathbb{E}_{c \sim Q} \left[ \ln \frac{Q(c)}{P(c)} \right]$$

with  $\hat{e}_Q$  and  $e_Q$  considered as distributions on  $\{0, +1\}$ .

We consider the following function class

$$\mathcal{F} = \left\{ f_{\mathbf{w}} : \mathbf{x} \in \mathcal{X} \mapsto \text{sgn} \left( \sum_{i=1}^N w_i x_i \right) : \|\mathbf{w}\|_1 \leq 1 \right\},$$

where we assume that  $\mathcal{X}$  is a subset of the  $\ell_\infty$  ball of radius 1, that is all components of  $\mathbf{x}$  in the support of the distribution have absolute value bounded by 1.

We are considering a frequentist style bound, so that we posit a fixed but unknown distribution  $\mathcal{D}$  that governs the generation of the input data, be it i.i.d. in the training set or as a test example. We would like to apply a margin based PAC-Bayes bound to such 1-norm regularised classifiers. For a given choice of weight vector  $\mathbf{w}$  with  $\|\mathbf{w}\|_1 = 1$ , for which we may expect many components to be equal to zero, we wish to create a posterior distribution  $Q(\mathbf{w})$  such that we can bound

$$P_{(\mathbf{x}, y) \sim \mathcal{D}} (f_{\mathbf{w}}(\mathbf{x}) \neq y) \leq 2e_{Q(\mathbf{w})},$$

where  $e_{Q(\mathbf{w})}$  is the expected error under the distribution  $Q(\mathbf{w})$ :

$$e_{Q(\mathbf{w})} = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}, q \sim Q(\mathbf{w})} [I [q(\mathbf{x}) \neq y]].$$

Given a training sample  $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ , we similarly define

$$\hat{e}_{Q(\mathbf{w})} = \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{q \sim Q(\mathbf{w})} [I [q(\mathbf{x}_i) \neq y_i]].$$

We first define the posterior distribution  $Q(\mathbf{w})$ . The general form of a classifier  $q$  will involve a random weight vector  $W \in \mathbb{R}^N$  together with a random threshold  $\Theta$  and the output will be

$$q_{W, \Theta}(\mathbf{x}) = \text{sgn}(\langle W, \mathbf{x} \rangle - \Theta).$$

The distribution  $Q(\mathbf{w})$  of  $W$  will be discrete with

$$W = \text{sgn}(w_i) \mathbf{e}_i; \text{ with probability } |w_i|, i = 1, \dots, N,$$

where  $\mathbf{e}_i$  is the unit vector with 1 in dimension  $i$  and zeros in all other dimensions. The distribution of  $\Theta$  is uniform on the interval  $[-1, 1]$ .

**Proposition 2.2** *With the above definitions, we have for  $\mathbf{w}$  satisfying  $\|\mathbf{w}\|_1 = 1$ , that for any  $(\mathbf{x}, y) \in \mathcal{X} \times \{-1, +1\}$ ,*

$$P_{q \sim Q(\mathbf{w})} (q(\mathbf{x}) \neq y) = 0.5(1 - y \langle \mathbf{w}, \mathbf{x} \rangle).$$

**Proof** Consider a fixed  $(\mathbf{x}, y)$ . We have

$$\begin{aligned} & P_{q \sim Q(\mathbf{w})} (q(\mathbf{x}) \neq y) \\ &= \sum_{i=1}^N |w_i| P_{\Theta} (\text{sgn}(\text{sgn}(w_i) \langle \mathbf{e}_i, \mathbf{x} \rangle - \Theta) \neq y) \\ &= \sum_{i=1}^N |w_i| P_{\Theta} (\text{sgn}(\text{sgn}(w_i) x_i - \Theta) \neq y) \\ &= 0.5 \sum_{i=1}^N |w_i| (1 - y \text{sgn}(w_i) x_i) \\ &= 0.5(1 - y \langle \mathbf{w}, \mathbf{x} \rangle), \end{aligned}$$

as required.

**Proposition 2.3** *With the above definitions, we have for  $\mathbf{w}$  satisfying  $\|\mathbf{w}\|_1 = 1$ , that*

$$P_{(\mathbf{x}, y) \sim \mathcal{D}} (f_{\mathbf{w}}(\mathbf{x}) \neq y) \leq 2e_{Q(\mathbf{w})}.$$

**Proof** By Proposition 2.2 we have

$$P_{q \sim Q(\mathbf{w})} (q(\mathbf{x}) \neq y) = 0.5(1 - y \langle \mathbf{w}, \mathbf{x} \rangle).$$

Hence, it follows that

$$\begin{aligned} P_{q \sim Q(\mathbf{w})} (q(\mathbf{x}) \neq y) &> 0.5 \\ &\Leftrightarrow \\ f_{\mathbf{w}}(\mathbf{x}) &\neq y. \end{aligned}$$

We can now estimate the error of the stochastic classifier

$$\begin{aligned} e_{Q(\mathbf{w})} &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}, q \sim Q(\mathbf{w})} [I[q(\mathbf{x}) \neq y]] \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} \mathbb{E}_{q \sim Q(\mathbf{w})} [I[q(\mathbf{x}) \neq y]] \\ &\geq 0.5 P_{(\mathbf{x}, y) \sim \mathcal{D}}(f_{\mathbf{w}}(\mathbf{x}) \neq y), \end{aligned}$$

as required.

We can now make the application of the PAC-Bayes bound to 1-norm regularised linear classifiers.

**Theorem 2.4** *Let  $\mathcal{X}$  be a subset of the  $\ell_\infty$  1-ball in  $\mathbb{R}^N$ , and let  $\mathcal{D}$  be a distribution with support  $\mathcal{X}$ . With probability at least  $1 - \delta$  over the draw of training sets of size  $m$ , we have for all  $\mathbf{w}$  satisfying  $\|\mathbf{w}\|_1 = 1$  that*

$$\text{KL}(\hat{e}_{Q(\mathbf{w})} \| e_{Q(\mathbf{w})}) \leq \frac{1}{m} \left[ \sum_{i=1}^N |w_i| \ln |w_i| + \ln(2N) + \ln((m+1)/\delta) \right]$$

**Proof** The result follows from an application of Proposition 2.1 by choosing the prior to be uniform on all of the vectors  $\pm \mathbf{e}_i, i = 1, \dots, N$ .

The implicit bound given by the KL divergence is not always easy to read, so we introduce the following notation for the ‘inversion’ of the KL divergence.

$$\text{KL}^{-1}(\hat{e}, A) = \max_e \{e : \text{KL}(\hat{e} \| e) \leq A\},$$

implying that  $\text{KL}^{-1}(\hat{e}, A)$  is the largest value satisfying

$$\text{KL}(\hat{e}, \text{KL}^{-1}(\hat{e}, A)) \leq A.$$

**Corollary 2.5** *Let  $\mathcal{X}$  be a subset of the  $\ell_\infty$  1-ball in  $\mathbb{R}^N$ , and let  $\mathcal{D}$  be a distribution with support  $\mathcal{X}$ . With probability at least  $1 - \delta$  over the draw of training sets of size  $m$ , we have for all  $\mathbf{w}$  satisfying  $\|\mathbf{w}\|_1 = 1$  that*

$$e_{Q(\mathbf{w})} \leq \text{KL}^{-1}(\hat{e}_{Q(\mathbf{w})}, H)$$

where

$$H = \frac{\sum_{i=1}^N |w_i| \ln(|w_i|) + \ln(2N) + \ln((m+1)/\delta)}{m}$$

The expression given by Proposition 2.2 for the empirical error is too weak to obtain a strong bound. We will therefore ‘boost’ the power of discrimination by sampling  $T$  copies of the distribution  $\mathbf{q} \sim Q^T(\mathbf{w})$  and then use the classification

$$q_{\mathbf{w}, \Theta}(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^T \text{sgn}(\langle W^t, \mathbf{x} \rangle - \Theta^t) \right), \quad (1)$$

or in other words taking a majority vote of the  $T$  samples to decide the classification given to the input  $\mathbf{x}$ . The effect on the KL divergence is a simple multiplication by the factor  $T$ , while the bound on the true error given by Proposition 2.3 remains valid. Hence, we arrive at the following further corollary.

**Corollary 2.6** *Let  $\mathcal{X}$  be a subset of the  $\ell_\infty$  1-ball in  $\mathbb{R}^N$ , and let  $\mathcal{D}$  be a distribution with support  $\mathcal{X}$ . With probability at least  $1 - \delta$  over the draw of training sets of size  $m$ , we have for all  $\mathbf{w}$  satisfying  $\|\mathbf{w}\|_1 = 1$  that*

$$P_{(\mathbf{x}, y) \sim \mathcal{D}}(f_{\mathbf{w}}(\mathbf{x}) \neq y) \leq 2\text{KL}^{-1}(\hat{e}_{Q^T(\mathbf{w})}, H),$$

where  $H = \frac{T \sum_{i=1}^N |w_i| \ln(|w_i|) + T \ln(2N) + \ln((m+1)/\delta)}{m}$  and  $\hat{e}_{Q^T(\mathbf{w})}$  is the empirical error of the classifier given by equation (1).

Finally, note that it is straightforward to compute  $\hat{e}_{Q^T(\mathbf{w})}$  using Proposition 2.2 as the following derivation shows:

$$\begin{aligned} \hat{e}_{Q^T(\mathbf{w})} &= \sum_{t=0}^{\lfloor T/2 \rfloor} \binom{T}{t} (1 - P(q(\mathbf{x}) \neq y))^t \\ &\quad \cdot P(q(\mathbf{x}) \neq y)^{T-t} \\ &= \sum_{t=0}^{\lfloor T/2 \rfloor} \binom{T}{t} (1 - 0.5(1 - y\langle \mathbf{w}, \mathbf{x} \rangle))^t \\ &\quad \cdot (0.5(1 - y\langle \mathbf{w}, \mathbf{x} \rangle))^{T-t} \\ &= 0.5^T \sum_{t=0}^{\lfloor T/2 \rfloor} \binom{T}{t} (1 + y\langle \mathbf{w}, \mathbf{x} \rangle)^t \\ &\quad \cdot (1 - y\langle \mathbf{w}, \mathbf{x} \rangle)^{T-t}. \end{aligned}$$

The function  $\hat{e}_{Q^T(\mathbf{w})}$  exhibits a sharper reverse sigmoid like behaviour as  $y\langle \mathbf{w}, \mathbf{x} \rangle$  increases and with  $T$  controlling the steepness of the cutoff.

### 3 ALGORITHMICS

The generalisation bound motivates the following ‘maximum entropy’ optimisation.

$$\begin{aligned} \min_{\mathbf{w}, \rho, \xi} \quad & \sum_{j=1}^N |w_j| \ln |w_j| - C\rho + D \sum_{i=1}^m \xi_i \\ \text{subject to:} \quad & y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq \rho - \xi_i, 1 \leq i \leq m, \\ & \|\mathbf{w}\|_1 \leq 1, \xi_i \geq 0, 1 \leq i \leq m. \end{aligned}$$

This section will investigate solution of this convex optimisation problem using duality methods. We will derive an algorithm that has been implemented in some small experiments in the next section.

Using the decomposition of  $w_j = w_j^+ - w_j^-$  with  $w_j^+, w_j^- \geq 0$ , we form the Langrangian

$$\begin{aligned} L = & \sum_{j=1}^N (w_j^+ + w_j^-) \ln(w_j^+ + w_j^-) - C\rho + D \sum_{i=1}^m \xi_i \\ & + \lambda \left( \sum_{j=1}^N (w_j^+ + w_j^-) - 1 \right) \\ & - \sum_{i=1}^m \alpha_i \left( y_i \sum_{j=1}^N (w_j^+ - w_j^-) x_{ij} - \rho + \xi_i \right) \\ & - \sum_{i=1}^m \xi_i \beta_i - \sum_{j=1}^N \eta_j^+ w_j^+ - \sum_{j=1}^N \eta_j^- w_j^- \end{aligned}$$

Taking derivatives with respect to the primary variables and setting equal to zero gives

$$\begin{aligned} \frac{\partial L}{\partial w_j^+} = & \ln(w_j^+ + w_j^-) + 1 - \sum_{i=1}^m \alpha_i y_i x_{ij} + \lambda \\ & - \eta_j^+ = 0, \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{\partial L}{\partial w_j^-} = & \ln(w_j^+ + w_j^-) + 1 + \sum_{i=1}^m \alpha_i y_i x_{ij} + \lambda \\ & - \eta_j^- = 0 \end{aligned} \quad (3)$$

$$\frac{\partial L}{\partial \xi_i} = D - \alpha_i - \beta_i = 0, \Rightarrow \alpha_i \leq D \quad (4)$$

$$\frac{\partial L}{\partial \rho} = -C + \sum_{i=1}^m \alpha_i = 0, \Rightarrow \sum_{i=1}^m \alpha_i = C. \quad (5)$$

Furthermore, adding equations (2) and (3) gives

$$w_j^+ + w_j^- = \exp\left(\frac{\eta_j^+ + \eta_j^-}{2} - 1 - \lambda\right), \quad (6)$$

while subtracting them gives

$$\sum_{i=1}^m \alpha_i y_i x_i = \frac{1}{2} (\eta^- - \eta^+). \quad (7)$$

Finally, summing over  $j$  and adding  $w_j^+$  times equation (2) plus  $w_j^-$  times equation (3) gives

$$\begin{aligned} & \sum_{j=1}^N (w_j^+ + w_j^-) \ln(w_j^+ + w_j^-) + \sum_{j=1}^N (w_j^+ + w_j^-) \\ & + \lambda \sum_{j=1}^N (w_j^+ + w_j^-) - \sum_{i=1}^m \alpha_i y_i \sum_{j=1}^N (w_j^+ - w_j^-) x_{ij} \\ & - \sum_{j=1}^N \eta_j^+ w_j^+ - \sum_{j=1}^N \eta_j^- w_j^- = 0. \end{aligned}$$

Subtracting this from the objective we obtain

$$L = - \sum_{j=1}^N (w_j^+ + w_j^-) - \lambda.$$

Using equation (6) we obtain the dual problem

$$\max_{\alpha, \eta^+, \eta^-} L = - \sum_{j=1}^N \exp\left(\frac{\eta_j^+ + \eta_j^-}{2} - 1 - \lambda\right) - \lambda$$

$$\text{subject to: } \sum_{i=1}^m \alpha_i y_i x_i = \frac{1}{2} (\eta^- - \eta^+),$$

$$\begin{aligned} \eta_j^+, \eta_j^- \geq 0, \quad 1 \leq j \leq N, \quad \sum_{i=1}^m \alpha_i = C, \\ 0 \leq \alpha_i \leq D, \quad 1 \leq i \leq m. \end{aligned}$$

Finally, using equation (7) we can eliminate  $\eta^+$  and  $\eta^-$  to obtain a simplified expression

$$\max_{\alpha} L = - \sum_{j=1}^N \exp\left(\left|\sum_{i=1}^m \alpha_i y_i x_{ij}\right| - 1 - \lambda\right) - \lambda$$

$$\text{subject to: } \sum_{i=1}^m \alpha_i = C \quad 0 \leq \alpha_i \leq D, 1 \leq i \leq m.$$

Taking a gradient ascent algorithm we can update  $\alpha$  along the gradient given by

$$\alpha \leftarrow \alpha + \zeta \left(\frac{\partial L}{\partial \alpha}\right) \quad (8)$$

where

$$\left(\frac{\partial L}{\partial \alpha}\right)_t = - \sum_{j=1}^N |w_j| \operatorname{sgn}\left(\sum_{i=1}^m \alpha_i y_i x_{ij}\right) y_t x_{tj} + \mu, \quad (9)$$

where we have introduced a Lagrange multiplier  $\mu$  for the 1-norm constraint on  $\alpha$  and

$$|w_j| = A \exp\left(\left|\sum_{i=1}^m \alpha_i y_i x_{ij}\right| - 1\right) \quad (10)$$

with  $A = \exp(-\lambda)$  chosen so that  $\sum_{j=1}^N |w_j| = 1$ . We should only involve those  $\alpha_i$  in the update for which  $0 < \alpha_i < D$ , or if  $\alpha_i = 0$  and the gradient is positive, or if  $\alpha_i = D$  and the gradient is negative. After the update with small learning rate  $\zeta$  we move each  $\alpha_i$  back into the interval  $[0, D]$  and update  $\mu$  by

$$\mu \leftarrow \mu - \tau \left(\sum_{i=1}^m \alpha_i - C\right), \quad (11)$$

for a smaller learning rate  $\tau$ . Note that by equation (7) and the definition of  $\eta^+$  and  $\eta^-$

$$\operatorname{sgn}(w_j) = \operatorname{sgn}\left(\sum_{i=1}^m \alpha_i y_i x_{ij}\right), \quad (12)$$

since  $w_j$  positive implies  $\eta_j^- - \eta_j^+ > 0$ .  
Hence, the resulting algorithm is given as follows.

**Algorithm** *Maximum Entropy Classification*

**Input:** Matrix of  $m$  training examples  $X$ , Parameters  $C, D$ .

**Output:** Vector of weights  $w$

1. Initialise  $\alpha_i = 1/C$  for all  $i$
2. Initialise  $\mu = 0$
3. Compute  $|w|$  using equation (10) with  $A$  chosen so 1-norm is 1.
4. Compute gradients using equation (9)
5.  $I = (1 : m)$
6. **repeat**
7.     **repeat**
8.         update  $\alpha(I)$  using equation (8)
9.         if  $\alpha_i \notin [0, D]$  remove  $i$  from  $I$  and set  $\alpha_i$  to 0 or  $D$ .
10.        Compute  $|w|$  using equation (10) with  $A$  chosen so 1-norm is 1.
11.        Compute gradients for indices  $I$  using equation (9)
12.     **until** no change
13.     Update  $\mu$  using equation (11)
14.     Compute all gradients using equation (9)
15.     Include in  $I$  any  $i$  for which  $\alpha_i = 0$  but gradient positive, or  $\alpha_i = D$  and gradient negative
16. **until** no change;
17. Use equation (12) to adjust the sign of  $w_j$ .

### 3.1 KERNEL MAXIMUM ENTROPY

We are able to extend the current framework to non-linear features through using the kernel trick and computing a Cholesky decomposition of the resulting kernel matrix.

$$\begin{aligned} K &= X'X \\ &= R'Q'QR \\ &= R'R \end{aligned}$$

The computation of  $R_{ij}$  corresponds to evaluating the inner product between  $\phi(x_i)$  with the new basis vector  $q_j$  for  $j < i$ . The basis vector  $q_j$  are the result of an implicit Gram-Schmidt orthonormalisation of the data in the feature space. We are able to view the new representation as a new projection function into a lower dimensional subspace. This new representation of the data in the columns of matrix  $R$ ,  $\mathbf{r}_i$ , which gives the exact same kernel matrix.

$$\hat{\phi} : \phi(\mathbf{x}_i) \rightarrow \mathbf{r}_i.$$

where  $\mathbf{r}_i$  is the  $i$ th column of  $R$ . The resulting kernel maximum entropy maximisation is

$$\max_{\alpha} \quad L = - \sum_{j=1}^N \exp \left( \left| \sum_{i=1}^m \alpha_i y_i r_i \right| - 1 - \lambda \right) - \lambda$$

$$\text{subject to:} \quad \sum_{i=1}^m \alpha_i = C \quad 0 \leq \alpha_i \leq D, 1 \leq i \leq m.$$

## 4 EXPERIMENTS

### 4.1 TESTING THE BOUND

Initially we test the algorithm and resulting bound on two of UCI datasets summarised in Table 1. We are interested in observing the behaviour of the bound for a given training and testing set in the linear case. In this following experiment we fix  $C = 1$  and  $D = \frac{1}{0.05\ell}$  where  $\ell$  is the number of samples.

Table 1: Datasets considered

Data	Features	Training	Testing
Ionosphere	34	250	101
Breast	9	487	196

The algorithm was run on each of these sets and the bound of Corollary 2.6 was computed for different values of  $T$ . Figure 1 shows a plot of the value of the bound for the Ionosphere data as a function of  $T$ . As indicated in Section 2,  $T$  controls the rate at which the values dip as we move away from the margin, but we pay a penalty in the multiplication of the KL divergence. Hence, its role is very similar to the margin parameter in SVM learning bounds. Note that we should introduce an extra penalty of  $\ln(40)/m$  as we have effectively applied the theorem 40 times for the different values of  $T$ , but we ignore this detail in our reported results. A plot for the Breast Cancer dataset is given in Figure 2.

Finally, Table 2 shows the test error with that given by the minimum bound (over  $T$ ) on these small datasets.

Table 2: Test errors and bound values

Data	Error	Bound
Ionosphere	0.22	0.94
Breast	0.02	0.51

While the bound values are far from non-trivial, they do show that the bound is able to deliver interesting values, particularly in the case of the Breast cancer dataset. Note that the factor of 2 has been included, something frequently omitted in reporting PAC-Bayes simulation results.

Table 3: Datasets description: Each row contains the name of the dataset, the number of samples and features (i.e. attributes) as well as the total number of positive and negative samples.

Dataset	# Samples	# Features	# Positive Samples	# Negative Samples
Votes	52	16	18	34
Glass	163	9	87	76
Haberman	294	3	219	75
Bupa	345	6	145	200
Credit	653	15	296	357
Pima	768	8	269	499
BreastW	683	9	239	444
Ionosphere	351	34	225	126

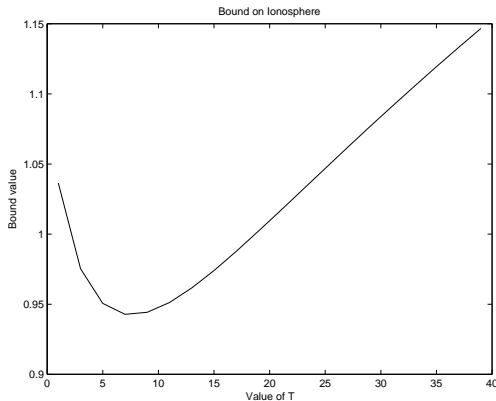


Figure 1: Plot of bound as a function of  $T$  for Ionosphere dataset

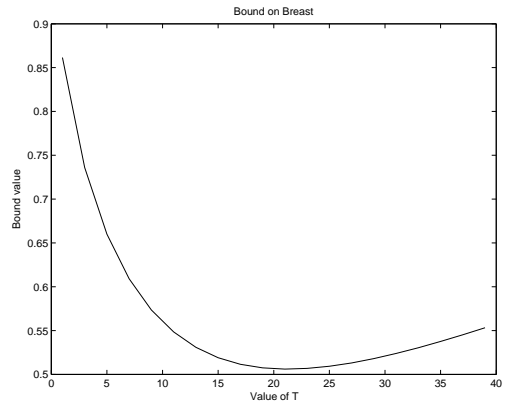


Figure 2: Plot of bound as a function of  $T$  for Breast Cancer dataset

#### 4.2 LINEAR & NONLINEAR RESULTS

We now test the algorithm in its linear and nonlinear variation on the eight UCI datasets summarised in Table 3. Furthermore, we are able to observe that the bound in Corollary 2.5 does not depend on  $C, D$  therefore allowing us to choose values that minimise the bound.

We first consider using the linear feature space and compare the maximum entropy algorithm with a Support Vector Machine (SVM) (with linear kernel) using cross-validation to determine the best  $C$  value, while the  $C$  and  $D$  for the maximum entropy method were chosen to optimise the bound. Table 4 gives the test errors for the two algorithms on the UCI datasets together with the bound value for the maximum entropy method. The algorithm has very similar quality on Votes, Bupa, and Pima. The maximum entropy method performs better on Glass, Haberman and Credit, while SVM is better on Ionosphere and significantly better on BreastW. Overall the results are encouraging though the BreastW results are concerning and require further investigation.

In the nonlinear case we compute the Gaussian kernel with a variety of values for the width parameter  $\gamma$ . For the maximum entropy method we first perform a complete Cholesky decomposition of the kernel matrix and use the obtained features as the representation of the data. The  $\gamma$  value was determined using cross-validation for both algorithms, with the parameters  $C$  for the SVM also determined by cross-validation, but  $C$  and  $D$  for the maximum entropy method chosen to optimise the bound. Table 5 shows the results for both algorithms together with the bound values for the maximum entropy method. In this case the two methods give similar results for Votes only, with the maximum entropy method performing slightly worse than the SVM in Glass and Bupa, but considerably worse on the other datasets.

The results for the non-linear datasets are disappointing and we speculate that this is because of the poor choice of representation using the Cholesky decomposition. There is a possibility of using the bound to prioritise the choice of features by selecting features



Table 4: Test errors and bound values for the linear max entropy algorithm and test errors for the SVM using a linear kernel. The  $C$  SVM parameter was selected using cross validation while the  $C, D$  max entropy parameters were selected by minimising the bound.

Data	SVM Error	Max Entropy Error	Max Entropy Bound
Votes	$0.3464 \pm 0.0113$	$0.3464 \pm 0.0113$	$1.5515 \pm 0.0047$
Glass	$0.4662 \pm 0.0056$	$0.4600 \pm 0.0144$	$1.3730 \pm 0.0003$
Haberman	$0.2551 \pm 0.0000$	$0.2517 \pm 0.0059$	$1.2742 \pm 0.0015$
Bupa	$0.4203 \pm 0.0026$	$0.4203 \pm 0.0026$	$1.2680 \pm 0.0006$
Credit	$0.2788 \pm 0.0679$	$0.2512 \pm 0.0072$	$1.2023 \pm 0.0007$
Pima	$0.3503 \pm 0.0013$	$0.3503 \pm 0.0013$	$1.1883 \pm 0.0002$
BreastW	$0.0658 \pm 0.0488$	$0.6501 \pm 0.0017$	$1.1959 \pm 0.0006$
Ionosphere	$0.2422 \pm 0.0261$	$0.2849 \pm 0.0471$	$1.2599 \pm 0.0006$

Table 5: Test errors and bound values for the non linear max entropy algorithm and test errors for the SVM using a Gaussian kernel. The  $C, \gamma$  SVM parameter was selected using cross validation while the  $C, D$  max entropy parameters were selected by minimising the bound, the Gaussian  $\gamma$  parameter was selected using cross-validation.

Data	SVM Error	Max Entropy Error	Max Entropy Bound
Votes	$0.0926 \pm 0.1604$	$0.0926 \pm 0.1604$	$1.5601 \pm 0.0026$
Glass	$0.2760 \pm 0.0159$	$0.2884 \pm 0.1488$	$1.3608 \pm 0.0142$
Haberman	$0.2483 \pm 0.0257$	$0.3129 \pm 0.0915$	$1.2044 \pm 0.0744$
Bupa	$0.2958 \pm 0.0200$	$0.3130 \pm 0.0127$	$1.2713 \pm 0.0005$
Credit	$0.2834 \pm 0.0674$	$0.3737 \pm 0.0449$	$1.2066 \pm 0.0001$
Pima	$0.2525 \pm 0.0242$	$0.2942 \pm 0.0257$	$1.1924 \pm 0.0001$
BreastW	$0.0366 \pm 0.0154$	$0.0483 \pm 0.0432$	$1.1996 \pm 0.0005$
Ionosphere	$0.0741 \pm 0.0300$	$0.1652 \pm 0.0486$	$1.2664 \pm 0.0030$

greedily that maximise the value

$$\left| \sum_{i=1}^m \alpha_i y_i x_{ij} \right|$$

that appears in the dual objective. This would correspond to the criterion used in boosting where the  $\alpha_i$  give a pseudo distribution over the examples in which the weak learner must give good correlation with the target. This could be used to drive a general boosting strategy for selecting weak learners from a large pool. This is, however, beyond the scope of this paper and will be the subject of further research.

## 5 CONCLUSIONS

The paper has developed new technology for applying PAC-Bayes analysis to an approach that has been of interest both in statistics and machine learning, namely the implementation of the principle of maximum entropy. Though developed only for classification in the current paper we believe that the extension to more complex tasks such as density modelling where the technique comes into its own will now be possible. In addition we have analysed the resulting convex op-

timisation problem that arises from the derived bound and shown how a dual version gives rise to a simple and efficient algorithmic implementation of the technique. It is an interesting feature of this algorithm that though the weight vector  $w$  is not sparse the dual variables  $\alpha$  are as one might expect from the resulting 1-norm constraint and as our experiments demonstrate.

Finally, we have implemented the algorithm on some UCI datasets and demonstrated that the factor  $T$  in the bound behaves in a similar manner to a margin parameter. We have use the bound to drive the model selection over the two parameters  $C$  and  $D$  of the algorithm and have demonstrated that the approach can be applied in kernel defined featured spaces using the Cholesky decomposition to generate an explicit representation. While the results for the linear feature representations are very encouraging those for the non-linear case are somewhat disappointing. We speculate that more care needs to be taken in the choice of representation in this case.

## Acknowledgements

The authors would like to acknowledge financial support from the EPSRC project Le Strum<sup>1</sup>, EP-D063612-1.

## References

- Dudik, M., & Schapire, R. E. (2006). Maximum entropy distribution estimation with generalized regularization. In *Proceedings of the conference on learning theory (colt)* (pp. 123–138).
- Kapur, J. N., & Kesevan, H. K. (1992). *Entropy optimization principles with applications*. Academic Press.
- Langford, J. (2005). Tutorial on practical prediction theory for classification. , *6*, 273–306.
- McAllester, D. A. (1998). Some PAC-Bayesian theorems. In *Proceedings of the 11th annual conference on computational learning theory* (pp. 230–234). ACM Press.
- McAllester, D. A. (1999). PAC-Bayesian model averaging. In *Proceedings of the 12th annual conference on computational learning theory* (pp. 164–170). ACM Press.
- Seeger, M. (2003). *Bayesian gaussian process models: Pac-bayesian generalisation error bounds and sparse approximations*. Unpublished doctoral dissertation, University of Edinburgh.
- Wang, S., Schuurmans, D., Peng, F., & Zhao, Y. (2004). Learning mixture models with the regularized maximum entropy principle. *IEEE Transactions on Neural Networks*, *15*(4), 903–916.

---

<sup>1</sup><http://www.lestrum.org>