

Chapter 3

PAC-Bayesian Bounds for Gaussian Process Methods

The PAC-Bayesian theorem of McAllester is an unusual result in the field of statistical learning theory. While other theorems employ heavy concepts from empirical process theory and beyond, the PAC-Bayesian theorem can be proved without much effort, using simple properties of convex functions (one of the contributions of this chapter is to give such a simple proof). Nevertheless, McAllester’s result is very powerful, applicable to a large number of Bayes-like learning schemes and highly configurable to available task prior knowledge, characteristics which most other PAC bounds lack to that extent. Maybe because of its simple and direct proof, the PAC-Bayesian theorem can also be very tight in practically relevant situations when many other “heavy-weight” theorems struggle to give non-trivial guarantees. In this chapter, we present various extensions of this remarkable result, along with a simple proof which points out convex (Legendre) duality as core principle, in contrast to most other recent PAC results which are based on the union bound and combinatorics. We then apply the theorem to approximate Bayesian Gaussian process classification, obtaining very tight bounds as judged by experimental studies. For fairly small sample sizes, the results are highly non-trivial and outperform other recent kernel classifier bounds we compared against by a wide margin.

The structure of this chapter is as follows. Section 3.1 is introductory and stresses the need in practice for bounds which are strongly dependent on learning method, training sample and task prior knowledge. In Section 3.2 we introduce several variants of the PAC-Bayesian theorem along with a proof and consider some extensions. These theorems are applied to Gaussian process classification in Section 3.3. In Section 3.4, we mention related work, and in Section 3.5 we present experimental results on a handwritten digits recognition task. The chapter is closed by the discussion in Section 3.6.

Parts of the results presented here have been published previously, as is de-

tailed in Section 1.1.

3.1 Data-dependent PAC Bounds and PAC-Bayesian Theorems

In this section, we discuss a number of shortcomings of classical Vapnik-Chervonenkis generalisation error bounds w.r.t. tightness in practical applications and show directions for improvement. We introduce a number of Bayesian types of classifiers. Finally, PAC-Bayesian theorems are motivated as a way to address the shortcomings of the classical results.

3.1.1 The Need for Data-dependent Bounds

Recall the binary classification problem introduced in Section A.5, where the training sample is $S = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$. Also recall the notion of a PAC bound on the generalisation error introduced in Section 2.2.1, which provides an estimated upper bound as a function of the training sample S and a confidence parameter δ , such that the event of the bound being violated has probability less than δ over random draws of S . Useful PAC bounds can only be proved in practically relevant situations if the complexity of the range of classification functions in response to finite training data is limited, e.g. via regularisation (see Section A.5). Without such limitation, the problem of learning from finite data is ill-posed.

The ideas behind classical VC bounds have been introduced in Section 2.2.3. It is important to note that the classical theory has not necessarily been developed with practical applications in mind, but rather to answer the question under which conditions the principle of empirical risk minimisation (ERM) over a hypothesis space \mathcal{H} is uniformly consistent. This question is solved completely by VC theory: \mathcal{H} is “learnable” iff its VC dimension is finite. Only much later have these results been linked with practically successful schemes such as the SVM (see Section 2.1.6). However, once applied to real-world situations, reasonable samples size and non-trivial models, classical VC bounds as well as many more recent developments have low or zero impact, because the bound values typically lie orders of magnitude above the truth.

Theoreticians tend to brush away such objections by citing minimax lower bounds which match the VC upper bounds fairly closely asymptotically. This argument holds up if we think of a PAC bound as a statement which is written down once and is then blindly and uniformly applied to any statistical problem at hand. But it is misleading in the context of real-world statistics, because the setup of these lower bounds is very different from the PAC scenario. Although a PAC bound holds for all data distributions, its value can depend strongly on

the data distribution. Modern bounds can be orders of magnitude smaller if the data distribution corresponds to the prior assumptions they encode than if it is constructed in a malicious way. Practitioners may not be too concerned with this issue, after all most of statistics and natural science would not work with nature as a malicious data distribution constructor! But exactly this is the setup of minimax lower bounds: given a particular statistical method, there are always some distributions which lead to very slow learning. However, these worst-case distributions are very unusual¹ and constructed in order to exploit weaknesses of the learning method.

Note that many researchers are more interested in bounding the rate of convergence of the gap as $n \rightarrow \infty$ than in precise gap bounds. The problem with this is that some insist the rate bounds have to be independent of the data distribution, thus to be the same for pathetic worst-case distributions than for any other more sensible distribution (in light of the model used). In order to obtain useful rate bounds in general (e.g., to show consistency) the model has to be restricted or regularised appropriately, and ideally theoretical results should guide these limiting choices in practice. But if a result does not depend on the true source, it will in general not support the single most important principle in statistics: obtain as much information as possible about the task and adjust the model to be compatible with this information. Rather, it will suggest to hedge against worst-case scenarios by choosing unrealistically simple models. In fact, it will typically be indifferent to whether we try to encode prior information faithfully or not, or even worse to vote against such efforts because they might increase our method’s vulnerability to worst-case scenarios.

Why is it possible to improve on classical VC results in practice? Recall that the *gap* is the difference between generalisation error and empirical error. A classical VC theorem will typically be based on a hypothesis space \mathcal{H} of finite VC dimension and will bound the gap for the worst choice of $h \in \mathcal{H}$. This bound will then of course hold for the particular algorithm we really use, but at the same time it holds just as well for any other method of choosing from \mathcal{H} , even for the “maximally malicious” algorithm which knows the data distribution perfectly and selects a hypothesis with maximal gap. This may be overly ambitious: in the words of Vapnik, *when solving a given problem, one should avoid solving a more general problem as an intermediate step* [199]. Sometimes it is possible to shift particularities of the algorithm into the definition of \mathcal{H} , but a better solution is to consider complexity measures other than the VC dimension (or scale-sensitive versions thereof) which are specific to the algorithm used. This can also alleviate another problem with classical VC bounds, namely that the gap bound depends on the sample S only via its size n . Recall from Section 2.2.1

¹They exhibit characteristics which typically make our model of them (on which a learning method is based) completely useless.

that such bounds are called data-independent or *a priori*, in contrast to data-dependent or *a posteriori* bounds. Since for any fixed predictor, the empirical error converges against the true one almost surely, the empirical error is certainly a major “ingredient” for any bound expression, but the limitation to this one and only statistic is sensible only in the classical setting where ERM alone is to be analysed. In the bounds we are interested in here, additional statistics are used to drive data-dependent complexity measures, potentially using more information in the sample than merely the empirical error and the size. Finally, classical VC bounds are restricted in how prior knowledge about the task might be encoded in the bound. This is possible to some degree, by creating a hierarchy² of nested hypothesis spaces of growing VC dimension corresponding to a prior based on Occam’s razor, but the process is very complicated in practice.

To summarise, the VC dimension of a hypothesis space seems unsuitable as a complexity measure in practice. It is neither flexible nor fine-grained enough and can hardly be adjusted to algorithms, models and prior knowledge. It also does not depend on the sample S . These shortcomings have been mentioned before, and the *luckiness framework* [178] has been proposed as an alternative. The PAC-Bayesian bounds to be discussed in this chapter can be seen as very strong realisations of this framework, but we will present them within the formally much simpler and more established framework of Bayesian inference. It is important to point out that we will not compromise the basic validity of PAC statements in any way:

- In order to construct a classification method *and* a data-dependent bound for it, we follow *Bayesian modelling assumptions* (or other heuristics): available prior knowledge is encoded, within feasibility constraints, into a probabilistic model and prior distributions, or an algorithm for prediction is derived in a different heuristic way. A distribution-free bound, which will in general depend on the algorithm, the prior assumptions and the sample S (beyond just the empirical error) is used to bound the generalisation error. The extent to which the unknown data distribution is compatible with these assumptions will in general determine the accuracy of the method *and* the observed tightness of the bound, but it does *not* compromise the validity of the theorem.
- The statement of the bound holds under standard *PAC assumptions*: We are given an i.i.d. training sample S from the data distribution which is otherwise *completely unknown*. Whether the data distribution is in agreement with the prior assumptions or violates them, does *not* influence the validity of the statement.

²This *structural risk minimisation* approach is used to deal with hypothesis spaces of infinite VC dimension, such as for example SVMs in a feature space.

3.1.2 Bayesian Classifiers. PAC-Bayesian Theorems

Recall the setup of the binary classification model defined in Section A.6.1, based on a family $\{u(\cdot|\mathbf{w})\}$ parameterised by \mathbf{w} and a noise distribution $P(y|u)$. We assume that the latter has the form $P(y|u) = f(y(u+b))$, where b is a bias hyperparameter, and f is symmetric around $(0, 1/2)$: $f(-x) = 1 - f(x)$. Note that \mathbf{w} need not be a finite-dimensional vector. For example, in our application to non-parametric models below we will identify \mathbf{w} with the process $u(\cdot|\mathbf{w})$ itself. A Bayesian analysis for this model (see Section A.6.2) requires the specification of a prior distribution $P(\mathbf{w})$. If $Q(\mathbf{w})$ is the posterior for the training sample S , the target probability for a new point \mathbf{x}_* is predicted as $Q(y_*|\mathbf{x}_*, S) = \mathbb{E}_Q[P(y_*|u(\mathbf{x}_*|\mathbf{w}))]$, and the *predictive classifier* is $\text{sgn}(Q(y_* = +1|\mathbf{x}_*, S) - 1/2)$.

A number of related classifiers have been studied. The *Bayes classifier* predicts $y_{\text{Bayes}}(\mathbf{x}_*) = \text{sgn}(\mathbb{E}_Q[u(\mathbf{x}_*|\mathbf{w})] + b)$, while the *Bayes voting classifier* outputs $y_{\text{Vote}}(\mathbf{x}_*) = \text{sgn} \mathbb{E}_Q[\text{sgn}(u(\mathbf{x}_*|\mathbf{w}) + b)]$. Note that our terminology is non-standard here: some authors would refer to our predictive classifier as Bayes classifier, while others use the term “Bayes classifier” to denote the optimal classifier for the data distribution. In general, all three classifiers (predictive, Bayes, Bayes voting) are different, but if the distribution of $u(\mathbf{x}_*|\mathbf{w})$, $\mathbf{w} \sim Q$ is symmetric around its mean for every \mathbf{x}_* , then they all agree. Another type of classifier, called *Gibbs classifier*, has been studied in learning theory (e.g., [70]). Given a test point \mathbf{x}_* , it predicts the corresponding target by first sampling $\mathbf{w} \sim Q$, then returning $y_* = \text{sgn}(u(\mathbf{x}_*|\mathbf{w}) + b)$, plugging in the sampled parameter vector. Note that a Gibbs classifier has a probabilistic element and requires coin tosses for prediction. Note also that if the targets of several test points are to be predicted, the parameter vectors sampled for this purpose are independent.³ In the situations we are interested in practice, the Gibbs classifier often performs somewhat worse than the corresponding Bayes classifier. We will discuss their relationship in more detail in Section 3.2.5. The Gibbs classifier is the method of choice if for some reason we are restricted to use a single $u(\mathbf{x}_*|\mathbf{w})$ for prediction.

In [116, 115, 117], McAllester proved a number of *PAC-Bayesian theorems* applicable to Gibbs classifiers. In general, a PAC-Bayesian theorem is simply a PAC bound which deals with Bayes-like classifiers constructed based on expectations over hypotheses or discriminants with respect to a posterior distribution Q . It is important to note that Q need not be a Bayesian posterior distribution for some model, but can be chosen by the learning algorithm at will. McAllester’s

³Readers familiar with *Markov chain Monte Carlo* methods (see Section A.6.3) will note the similarity with a MCMC approximation (based on one sample of \mathbf{w} only) of the corresponding Bayes classifier for $Q(\mathbf{w})$. The difference is that typically in MCMC, the sample representing the posterior $Q(\mathbf{w})$ is retained and used for many predictions, while in the Gibbs classifier, we use each posterior sample only once.

theorem incorporates all directions of improvement mentioned in Section 3.1.1 and suggests a new complexity measure which is data and algorithm-dependent and can be adjusted based on prior knowledge. More importantly, the measure is compatible with the aims of Bayesian modelling and prior assessment, so that the theorem is especially suitable for applications to (approximate) Bayesian algorithms. In the following section, we will present the theorem and a range of extensions, together with a simple and intuitive proof.

3.2 The PAC-Bayesian Theorem for Gibbs Classifiers

In this section, we present and prove a range of PAC-Bayesian theorems for Gibbs classifiers, both for the binary classification problem and more general multi-class scenarios or arbitrary bounded loss functions. A simple extension to binary Bayes classifiers is motivated as well.

3.2.1 The Binary Classification Case

In this section, we present McAllester’s PAC-Bayesian theorem for binary classification. The theorem deals with a Gibbs classifier (see Section 3.1.2) whose mixture distribution $Q(\mathbf{w})$ may depend on the training sample S , which is why $Q(\mathbf{w})$ is referred to as “posterior distribution”. In order to eliminate the probabilistic element in the Gibbs classifier itself, the bound is on the gap between expected generalisation error and expected empirical error, where the expectation is over $Q(\mathbf{w})$. The theorem can be configured by a prior distribution $P(\mathbf{w})$ over parameter vectors. The gap bound term depends strongly on the relative entropy (Definition A.4) $D[Q \parallel P]$ between $Q(\mathbf{w})$ and the prior $P(\mathbf{w})$. Here, we assume that $Q(\mathbf{w})$ and $P(\mathbf{w})$ are absolutely continuous w.r.t. some positive measure. Recall the Bernoulli relative entropy $D_{\text{Ber}}[q \parallel p]$ from (A.7). It is convex in (q, p) , furthermore $D_{\text{Ber}}[q \parallel \cdot]$ is strictly decreasing for $p < q$, strictly increasing for $p > q$. Thus, the following mapping

$$D_{\text{Ber}}^{-1}(q, \varepsilon) = [p_L, p_U] \text{ s.t. } D_{\text{Ber}}[q \parallel p_L] = D_{\text{Ber}}[q \parallel p_U] = \varepsilon, \quad p_L \leq q, \quad p_U \geq q, \quad (3.1)$$

is well-defined for $q \in (0, 1)$ and $\varepsilon \geq 0$. We also define $D_{\text{Ber}}^{-1}(0, \varepsilon) = [0, 1 - e^{-\varepsilon}]$ and $D_{\text{Ber}}^{-1}(q, \infty) = [0, 1]$. $D_{\text{Ber}}^{-1}(q, \varepsilon)$ can be seen as “relative entropy ball” of radius ε around q . Note that due to the convexity of D_{Ber} , we can compute the interval limits of $D_{\text{Ber}}^{-1}(q, \varepsilon)$ easily using Newton’s algorithm. It is clear by definition that for $\varepsilon \geq 0$, $p \in [0, 1]$:

$$p \in D_{\text{Ber}}^{-1}(q, \varepsilon) \iff D_{\text{Ber}}[q \parallel p] \leq \varepsilon. \quad (3.2)$$

If $\delta \in (0, 1)$ is a confidence parameter, we have the following result.

Theorem 3.1 (PAC-Bayesian theorem [115]) *For any data distribution over $\mathcal{X} \times \{-1, +1\}$, we have that the following bound holds, where the probability is over random i.i.d. samples $S = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$ of size n drawn from the data distribution:*

$$\Pr_S \left\{ \text{gen}(Q) \in D_{\text{Ber}}^{-1}(\text{emp}(S, Q), \varepsilon(\delta, n, P, Q)) \text{ for all } Q \right\} \geq 1 - \delta. \quad (3.3)$$

Here, $Q = Q(\mathbf{w})$ is an arbitrary “posterior” distribution over parameter vectors, which may depend on the sample S and on the prior P . Furthermore,

$$\begin{aligned} \text{emp}(S, Q) &= \mathbb{E}_{\mathbf{w} \sim Q} \left[\frac{1}{n} \sum_{i=1}^n \mathbf{I}_{\{\text{sgn}(u(\mathbf{x}_i | \mathbf{w}) + b) \neq y_i\}} \right], \\ \text{gen}(Q) &= \mathbb{E}_{\mathbf{w} \sim Q} \left[\mathbb{E}_{(\mathbf{x}_*, y_*)} \left[\mathbf{I}_{\{\text{sgn}(u(\mathbf{x}_* | \mathbf{w}) + b) \neq y_*\}} \right] \right], \\ \varepsilon(\delta, n, P, Q) &= \frac{1}{n} \left(D[Q \| P] + \log \frac{n+1}{\delta} \right). \end{aligned}$$

Here, $\text{emp}(S, Q)$ is the expected empirical error, $\text{gen}(Q)$ the expected generalisation error of the Gibbs classifier based on $Q(\mathbf{w})$ (note that the probability in $\text{gen}(Q)$ is over (\mathbf{x}_*, y_*) drawn from the data distribution, independently of the sample S).

Although D_{Ber}^{-1} is easy to compute, it may be awkward to use in certain situations. Therefore, D_{Ber} is frequently approximated by lower bounds which are fairly tight if $q \approx p$. If $p \geq q$, we have $D_{\text{Ber}}[q \| p] \geq (p - q)^2 / (2p)$ which can be seen by taking derivatives of both sides w.r.t. q . It follows that if $p \geq q$ and $D_{\text{Ber}}[q \| p] \leq \varepsilon$, then

$$p \leq q + 2\varepsilon + \sqrt{2\varepsilon q},$$

leading to

$$\text{gen}(Q) \leq \text{emp}(S, Q) + 2\varepsilon + \sqrt{2\varepsilon \text{emp}(S, Q)},$$

which shows that the gap bound scales roughly as $2D[Q \| P]/n$ if the empirical error is small. Needless to say, the use of this additional lower bound is not recommended in practice.

Note that McAllester’s theorem applies more generally to bounded loss functions and makes use of Hoeffding’s inequality for bounded variables. A further generalisation is shown in Section 3.2.4. However, for the special case of zero-one loss, we can use techniques tailored for binomial variables which give considerably tighter results than Hoeffding’s bound if the expected empirical error of the Gibbs classifier is small. The theorem can be generalised in various ways. In Section 3.2.2, we present a multi-class version which encompasses the binary classification case, and the proof given there will serve to prove Theorem 3.1. A slightly shorter direct proof can be found in [171]. Complexity measures other

than the relative entropy may be considered, as discussed in Section 3.2.6.1. Finally, a simple but rather weak extension to the Bayes classifier is given in Section 3.2.5.

We should stress once more, in line with Section 3.1.1, that the PAC-Bayesian theorem does *not* require the true data distribution to be constrained in any way depending on the prior $P(\mathbf{w})$ and the model class. For example, other analyses (which are not PAC) try to characterise *learning curves*: given that S has been generated from the given model and prior, they analyse the generalisation error of a Gibbs or Bayes classifier, averaged over the data distribution (e.g., [71, 187]). Or, given that S has been generated i.i.d. from a fixed member \mathbf{w}_0 of the family, they derive the convergence rate of some distance between this (product) data distribution and the Bayesian marginal likelihood $E_{P(\mathbf{w})}[P(S|\mathbf{w})]$ [30]. It is clear that under such more restrictive assumptions to start with, stronger results can be achieved than the PAC-Bayesian theorem.

3.2.2 Confusion Distributions and Multiple Classes

In practice, many classification problems come with more than two classes. Although we can always tackle such problems using a sufficient number of binary classifiers, it is more principled and data-economic to instead use a multi-class model (see Section 2.1.2). In this subsection, we assume that the targets to be predicted are class labels from $\{1, \dots, C\}$, $C \geq 2$. Probabilistic rules mapping input points \mathbf{x} to distributions over $\{1, \dots, C\}$ will also be considered. A rule $\pi(\cdot|\mathbf{w})$ is used to predict y_* at a test point \mathbf{x}_* by sampling it from the distribution $\pi(\mathbf{x}_*|\mathbf{w})$ over $\{1, \dots, C\}$. By restricting ourselves to delta distributions $\pi(\cdot|\mathbf{w})$, we can always recover the special case of purely deterministic rules.

The application of a PAC bound is really only a statistical test and as such has to be embedded into the experimental design. Even in the binary case ($C = 2$), we might be interested in bounding other aspects of the data distribution than the generalisation error, e.g. the probability of false positives, and as C grows so does the number of possible questions which can be tested based on the training sample. Especially in the multi-class case, a useful PAC bound should support individual designs concerned with more general properties than the generalisation error, including the latter as a special case. It is not hard to extend the PAC-Bayesian theorem in this respect. We capture the notion of probabilistic rules by introducing a variable $\mathbf{r} \sim R(\mathbf{r})$. \mathbf{r} is the source of random coins required to sample $y_* \sim \pi(\mathbf{x}_*|\mathbf{w})$. It is independent of all other variables, and R is fixed and independent of all other distributions. Effectively, $\pi_{y_*}(\mathbf{x}_*|\mathbf{w}) = \Pr_{\mathbf{r} \sim R}\{y(\mathbf{x}_*|\mathbf{w}, \mathbf{r}) = y_*\}$ for deterministic rules $y(\mathbf{x}|\mathbf{w}, \mathbf{r})$ mapping into $\{1, \dots, C\}$. Note that the corresponding Gibbs rule based on the distribution $Q(\mathbf{w})$ is evaluated at \mathbf{x}_* by sampling $\mathbf{w} \sim Q$, $\mathbf{r} \sim R$ independently, then outputting $y(\mathbf{x}_*|\mathbf{w}, \mathbf{r})$. The experimental design now implies a finite set \mathcal{L} of

size L and a distribution \mathbf{p} over \mathcal{L} which is related to the unknown data distribution and the posterior distribution $Q(\mathbf{w})$ as follows: if (\mathbf{x}_*, y_*) is sampled from the data distribution, $\mathbf{w} \sim Q$ and $\mathbf{r} \sim R$ independently, then the variable $l(\mathbf{x}_*, y_*, \mathbf{w}, \mathbf{r})$ has distribution \mathbf{p} , where l is a known function. For example, if $\mathcal{L} = \{0, 1\}$ and $l(\mathbf{x}_*, y_*, \mathbf{w}, \mathbf{r}) = \mathbb{I}_{\{y(\mathbf{x}_*|\mathbf{w}, \mathbf{r}) \neq y_*\}}$, then $\mathbf{p} = (1 - e_{\text{Gibbs}}, e_{\text{Gibbs}})$, where $e_{\text{Gibbs}} = \Pr\{y(\mathbf{x}_*|\mathbf{w}, \mathbf{r}) \neq y_*\}$ is the expected generalisation error of the Gibbs rule. More generally, we may be interested in the *joint confusion distribution*

$$F(y_*, \tilde{y}) = \mathbb{E}_{\mathbf{x}_*} [P(y_*|\mathbf{x}_*) \mathbb{E}_{\mathbf{w} \sim Q} [\pi_{\tilde{y}}(\mathbf{x}_*|\mathbf{w})]], \quad (3.4)$$

where $P(y_*|\mathbf{x}_*)$ denotes the conditional data distribution, for example in order to bound the probabilities of false positives and true negatives in the case of binary classification. Now, if $\mathcal{L} = \{1, \dots, C\} \times \{1, \dots, C\}$ and $l(\mathbf{x}_*, y_*, \mathbf{w}, \mathbf{r}) = (y_*, y(\mathbf{x}_*|\mathbf{w}, \mathbf{r}))$, then \mathbf{p} coincides with the joint confusion distribution F . In short, a general PAC bound allows us to make inferences about the components of \mathbf{p} based on the observed sample S .

We can now state the following generalisation of the PAC-Bayesian Theorem 3.1 for binary classification. Suppose we are given a set \mathcal{L} of size L and a mapping $l(\mathbf{x}_*, y_*, \mathbf{w}, \mathbf{r})$ into \mathcal{L} , furthermore an arbitrary prior distribution $P(\mathbf{w})$ over parameter vectors, and we choose a confidence parameter $\delta \in (0, 1)$. Then, the following result holds.

Theorem 3.2 (Extended PAC-Bayesian Theorem) *For any data distribution over $\mathcal{X} \times \{-1, +1\}$, we have that the following bound holds, where the probability is over random i.i.d. samples $S = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$ of size n drawn from the data distribution:*

$$\Pr_S \{D[\hat{\mathbf{p}}(S, Q) \parallel \mathbf{p}(Q)] \leq \varepsilon(\delta, n, P, Q) \text{ for all } Q\} > 1 - \delta. \quad (3.5)$$

Here, $Q = Q(\mathbf{w})$ is an arbitrary “posterior” distribution over parameter vectors, which may depend on the sample S and on the prior P . Furthermore, $\mathbf{p}(Q)$ is a distribution over \mathcal{L} , induced from the data distribution as follows:

$$[\mathbf{p}(Q)]_{l_*} = \Pr_{(\mathbf{x}_*, y_*), \mathbf{w}, \mathbf{r}} \{l(\mathbf{x}_*, y_*, \mathbf{w}, \mathbf{r}) = l_*\},$$

where (\mathbf{x}_*, y_*) is drawn from the data distribution, $\mathbf{w} \sim Q$ and $\mathbf{r} \sim R$, all independently. $\hat{\mathbf{p}}(S, Q)$ is an empirical estimate of $\mathbf{p}(Q)$ given by

$$[\hat{\mathbf{p}}(S, Q)]_{l_*} = \frac{1}{n} \sum_{i=1}^n \Pr_{\mathbf{w}, \mathbf{r}} \{l(\mathbf{x}_i, y_i, \mathbf{w}, \mathbf{r}) = l_*\}.$$

Furthermore,

$$\varepsilon(\delta, n, P, Q) = \frac{D[Q \parallel P] + (L - 1) \log(n + 1) - \log \delta}{n}.$$

The proof of this theorem is given below in this section. Note that Theorem 3.1 is a special case of Theorem 3.2: it is obtained if we set $\mathcal{L} = \{0, 1\}$ and $l(\mathbf{x}_*, y_*, \mathbf{w}, \mathbf{r}) = \mathbb{I}_{\{y(\mathbf{x}_*|\mathbf{w}, \mathbf{r}) \neq y_*\}}$ with $y(\mathbf{x}_*|\mathbf{w}, \mathbf{r}) = \text{sgn}(u(\mathbf{x}_*|\mathbf{w}) + b)$.

The theorem renders a level ε such that with high confidence $1 - \delta$ we have $D[\hat{\mathbf{p}} \parallel \mathbf{p}] \leq \varepsilon$, where $\hat{\mathbf{p}} = \hat{\mathbf{p}}(S, Q)$, $\mathbf{p} = \mathbf{p}(Q)$ and $\varepsilon = \varepsilon(\delta, n, P, Q)$. In other words, the unknown vector $\mathbf{p} \in \mathbb{R}^L$ lies in the closed convex set

$$\mathcal{P}(\hat{\mathbf{p}}, \varepsilon) = \left\{ \mathbf{q} \mid \mathbf{q} \geq 0, \mathbf{1}^T \mathbf{q} = 1, D[\hat{\mathbf{p}} \parallel \mathbf{q}] \leq \varepsilon \right\}. \quad (3.6)$$

$\mathcal{P}(\hat{\mathbf{p}}, \varepsilon)$ can be seen as “relative entropy ball” of radius ε around the centre $\hat{\mathbf{p}}$, a multidimensional generalisation of D_{Ber}^{-1} (see Equation 3.1). By cutting this ball with planes, we can derive bounds on projections $\mathbf{c}^T \mathbf{p}$. In Appendix B.1 we provide an explicit example for how the theorem can be used in practice.

As one of the major contributions of this chapter, we present a proof of Theorem 3.2. McAllester’s theorem [117] is a special case of this theorem, yet our proof is considerably simpler than the original one and leads to several possible avenues of generalisation. Our bound is also tighter in the special case of binary classification.

Recall the notation introduced further above in this section. For notational simplicity, we define $\tilde{\mathbf{w}} = (\mathbf{w}, \mathbf{r})$, thus pairing the two possible random sources of the randomised Gibbs rule.⁴ We also extend prior P and posterior Q by setting $dP(\tilde{\mathbf{w}}) = dP(\mathbf{w}) dR(\mathbf{r})$, $dQ(\tilde{\mathbf{w}}) = dQ(\mathbf{w}) dR(\mathbf{r})$ (product measures). Define

$$[\mathbf{p}(\tilde{\mathbf{w}})]_{l_*} = \mathbb{E}_{(\mathbf{x}_*, y_*)} [\mathbb{I}_{\{l(\mathbf{x}_*, y_*, \tilde{\mathbf{w}}) = l_*\}}], \quad [\hat{\mathbf{p}}(\tilde{\mathbf{w}})]_{l_*} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{\{l(\mathbf{x}_i, y_i, \tilde{\mathbf{w}}) = l_*\}}, \quad l_* \in \mathcal{L},$$

where the expectation is over the unknown data distribution, and the sample is $S = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$. Let $\Delta(\tilde{\mathbf{w}}) = D[\hat{\mathbf{p}}(\tilde{\mathbf{w}}) \parallel \mathbf{p}(\tilde{\mathbf{w}})]$. $\Delta(\tilde{\mathbf{w}})$ simply measures, for a fixed instance $\tilde{\mathbf{w}}$ of a fixed rule, the divergence between $\mathbf{p}(\tilde{\mathbf{w}})$ and its empirical estimate $\hat{\mathbf{p}}(\tilde{\mathbf{w}})$ in a convenient way.

Fix $\tilde{\mathbf{w}}$, and write $\hat{\mathbf{p}} = \hat{\mathbf{p}}(\tilde{\mathbf{w}})$, $\mathbf{p} = \mathbf{p}(\tilde{\mathbf{w}})$. Since $\tilde{\mathbf{w}}$ is fixed independent of the sample S , the strong law of large numbers asserts that $\hat{\mathbf{p}}$ converges against \mathbf{p} almost surely and we can derive a strong large deviation inequality for $\Delta(\tilde{\mathbf{w}})$. We may expect that this guarantee remains valid if instead of fixing $\tilde{\mathbf{w}}$ *a priori*, we sample it from the prior distribution $P(\tilde{\mathbf{w}})$, because P does not depend on the sample S either. The first part of the proof makes this argument sound. The reason for our *particular* choice of $\Delta(\tilde{\mathbf{w}})$ is that the corresponding large deviation inequality is tight and can be proved easily. For fixed $\tilde{\mathbf{w}}$, $n \hat{\mathbf{p}}$ is multinomial (n, \mathbf{p}) distributed. Csiszár and Körner [42] refer to $\hat{\mathbf{p}}$ as the *type* of the underlying i.i.d. sequence $\{l(\mathbf{x}_i, y_i, \tilde{\mathbf{w}}) \mid i = 1, \dots, n\}$, and we will use their elegant method of

⁴If the rules (parameterised by \mathbf{w}) are deterministic, we set $\tilde{\mathbf{w}} = \mathbf{w}$ and forget about \mathbf{r} altogether.

types (see also [34], Sect. 12.1) for the first part of the proof. As is shown in Appendix B.2, we have

$$\mathbb{E}_S [e^{nD[\hat{\mathbf{p}} \parallel \mathbf{p}]}] \leq (n+1)^{L-1}. \quad (3.7)$$

Now, taking the average over $\tilde{\mathbf{w}} \sim P$ and using Markov's inequality (Theorem A.1), we obtain

$$\Pr_S \left\{ \mathbb{E}_{\tilde{\mathbf{w}} \sim P} [e^{n\Delta(\tilde{\mathbf{w}})}] > \frac{(n+1)^{L-1}}{\delta} \right\} \leq \delta. \quad (3.8)$$

It is of course essential here that P does not depend on the sample S . In other words, it is easy to prove strong large deviation bounds for “dumb” Gibbs rules which do not depend on the training sample! For example, we can use the concavity of log and the convexity of $\Delta(\tilde{\mathbf{w}})$ together with Jensen's inequality (Lemma A.4) to see that

$$D \left[\mathbb{E}_{\tilde{\mathbf{w}} \sim P} [\hat{\mathbf{p}}(\tilde{\mathbf{w}})] \parallel \mathbb{E}_{\tilde{\mathbf{w}} \sim P} [\mathbf{p}(\tilde{\mathbf{w}})] \right] \leq \frac{(L-1) \log(n+1) - \log \delta}{n} \quad (3.9)$$

with probability at least $1 - \delta$ over random draws of S . Unfortunately, this is also quite uninteresting in practice. The cornerstone of the PAC-Bayesian theorem is a *generic* way of converting such bounds on “dumb” *a priori* Gibbs rules into useful bounds on *a posteriori* rules (the same method can be applied to Bayes rules as well, see Section 3.4.1).

Let us take the statement (3.8) for the “dumb” Gibbs rule based on the prior P and ask what happens if we replace P against the posterior Q . Fix an arbitrary sample S for which indeed

$$\mathbb{E}_{\tilde{\mathbf{w}} \sim P} [e^{n\Delta(\tilde{\mathbf{w}})}] \leq K, \quad K = \frac{(n+1)^{L-1}}{\delta}. \quad (3.10)$$

If we can show that

$$\mathbb{E}_{\tilde{\mathbf{w}} \sim Q} [n\Delta(\tilde{\mathbf{w}})] \leq D[Q \parallel P] + \log \mathbb{E}_{\tilde{\mathbf{w}} \sim P} [e^{n\Delta(\tilde{\mathbf{w}})}], \quad (3.11)$$

then we have that

$$\mathbb{E}_{\tilde{\mathbf{w}} \sim Q} [\Delta(\tilde{\mathbf{w}})] \leq \frac{D[Q \parallel P] + \log K}{n}. \quad (3.12)$$

Note that the relative entropy $D[Q \parallel P]$ between $Q(\tilde{\mathbf{w}})$ and $P(\tilde{\mathbf{w}})$ is identical to the relative entropy between $Q(\mathbf{w})$ and $P(\mathbf{w})$, because the factor $R(\mathbf{r})$ cancels out in the Radon-Nikodym derivative (recall Definition A.4).

We use the notion of convex (Legendre) duality (see Section A.3) to prove (3.11). $D[Q \parallel P]$ is convex in Q , and its convex dual is given by the log partition function $\log \mathbb{E}_P[\exp(\lambda(\tilde{\mathbf{w}}))]$ (see Equation A.8; \mathbf{w} has to be replaced by $\tilde{\mathbf{w}}$), where

the dual parameter $\lambda(\tilde{\mathbf{w}})$ is measurable w.r.t. $dP(\tilde{\mathbf{w}})$. To see this, we only have to consider $\lambda(\tilde{\mathbf{w}})$ such that $\mathbb{E}_P[\exp(\lambda(\tilde{\mathbf{w}}))] < \infty$. For such a candidate, define the Gibbs measure

$$dP_G(\tilde{\mathbf{w}}) = \frac{e^{\lambda(\tilde{\mathbf{w}})}}{\mathbb{E}_P[e^{\lambda(\tilde{\mathbf{w}})}]} dP(\tilde{\mathbf{w}}), \quad (3.13)$$

which is a probability measure relative to $P(\tilde{\mathbf{w}})$. The relative entropy is non-negative (see Section A.3), therefore

$$\begin{aligned} 0 \leq D[Q \parallel P_G] &= \int \log \left(\frac{\mathbb{E}_P[e^{\lambda(\tilde{\mathbf{w}})}]}{e^{\lambda(\tilde{\mathbf{w}})}} \frac{dQ(\tilde{\mathbf{w}})}{dP(\tilde{\mathbf{w}})} \right) dQ(\tilde{\mathbf{w}}) \\ &= D[Q \parallel P] + \log \mathbb{E}_P[e^{\lambda(\tilde{\mathbf{w}})}] - \mathbb{E}_Q[\lambda(\tilde{\mathbf{w}})]. \end{aligned} \quad (3.14)$$

Furthermore, if $D[Q \parallel P]$ is finite, then the density dQ/dP exists, and the inequality becomes an equality for $\lambda(\tilde{\mathbf{w}}) = \log(dQ(\tilde{\mathbf{w}})/dP(\tilde{\mathbf{w}})) + c$ for any c . Now, equation (3.11) follows from (A.8) if we use $\lambda(\tilde{\mathbf{w}}) = n \Delta(\tilde{\mathbf{w}})$.

We can conclude the proof by noting the convexity of the relative entropy (see Section A.3) and using Jensen's inequality. Namely, if (3.12) holds for S , then

$$\begin{aligned} D[\mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\hat{\mathbf{p}}(\tilde{\mathbf{w}})] \parallel \mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\mathbf{p}(\tilde{\mathbf{w}})]] &\leq \mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[D[\hat{\mathbf{p}}(\tilde{\mathbf{w}}) \parallel \mathbf{p}(\tilde{\mathbf{w}})]] \\ &\leq \frac{D[Q \parallel P] + (L-1) \log(n+1) - \log \delta}{n}. \end{aligned} \quad (3.15)$$

If we compare this inequality with the inequality (3.9) for the “dumb” classifier based on P , we see that we have to pay a penalty $n^{-1}D[Q \parallel P]$ for replacing the prior P by the posterior Q . Altogether, since $\hat{\mathbf{p}}(S, Q) = \mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\hat{\mathbf{p}}(\tilde{\mathbf{w}})]$, $\mathbf{p}(Q) = \mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\mathbf{p}(\tilde{\mathbf{w}})]$, we can combine (3.8) and the fact that for fixed S , (3.10) implies (3.15) in order to conclude that (3.5) must hold.

Theorem 3.1 is a special case of Theorem 3.2, and is obtained by setting $\mathcal{L} = \{0, 1\}$ and $l(\mathbf{x}_*, y_*, \mathbf{w}, \mathbf{r}) = \mathbb{I}_{\{y(\mathbf{x}_* | \mathbf{w}, \mathbf{r}) \neq y_*\}}$. Then, it is easy to see that $\mathbf{p}(Q) = (1 - \text{gen}(Q), \text{gen}(Q))$, $\hat{\mathbf{p}}(S, Q) = (1 - \text{emp}(S, Q), \text{emp}(S, Q))$ and $D[\hat{\mathbf{p}}(S, Q) \parallel \mathbf{p}(Q)] = D_{\text{Ber}}[\text{emp}(S, Q) \parallel \text{gen}(Q)]$. Theorem 3.1 follows from using (3.2).

3.2.3 Comments

We have seen above in Section 3.2.2 that the proof of the PAC-Bayesian theorem naturally decomposes into two parts. The first part is specific to the setting and consists of proving a large deviation inequality of the style (3.8) for a “dumb” Gibbs classifier which selects its rule based on the prior P , independent of the sample S . The second part is generic and quantifies the slack in this inequality that we have to pay if we want to replace the “dumb” classifier based on P against the Gibbs rule of interested, based on the posterior Q . This slack is quantified directly by the relative entropy $D[Q \parallel P]$ between posterior and prior.

Thus, whenever our learning method requires to select a posterior distribution Q which is concentrated on a region deemed very unlikely under P , a high cost has to be paid in the bound.

The prior distribution P seems to enter the PAC-Bayesian theorem out of “thin air”, yet its role is central as a parameter which can be tuned *a priori* to potentially tighten the bound.⁵ The choice of P is of course constrained by the requirement of independence from the sample S . Given that, it should be chosen to give rise to small $D[Q \parallel P]$ for samples S which we believe are likely to be observed for our task. If the PAC-Bayesian theorem is applied to an (approximate) Bayesian technique, this coincides with the typical Bayesian objective of coding available task prior knowledge into the prior distribution and the model class.

It is interesting to compare the terms in the PAC-Bayesian theorem with a frequently used Bayesian model selection criterion: the *log marginal likelihood*

$$\log P(\mathbf{y}) = \log \int P(\mathbf{y}|\mathbf{w}) dP(\mathbf{w})$$

(see Sections A.6.2 and 2.1.3). If we employ the true posterior in the PAC-Bayesian theorem, i.e. $Q(\mathbf{w}) = P(\mathbf{w}|\mathbf{y})$, we have

$$\log P(\mathbf{y}) = \mathbb{E}_{\mathbf{w} \sim Q} \left[\log \frac{P(\mathbf{y}|\mathbf{w}) dP(\mathbf{w})}{dP(\mathbf{w}|\mathbf{y})} \right] = \mathbb{E}_{\mathbf{w} \sim Q} [\log P(\mathbf{y}|\mathbf{w})] - D[Q \parallel P], \quad (3.16)$$

therefore

$$-\frac{1}{n} \log P(\mathbf{y}) = \frac{D[Q \parallel P]}{n} + \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\mathbf{w} \sim Q} [-\log P(y_i|\mathbf{x}_i, \mathbf{w})].$$

Thus, the normalised negative log marginal likelihood which is minimised in Bayesian model selection, is the sum of the complexity term employed in the PAC-Bayesian theorem and the sample average of $\mathbb{E}_{\mathbf{w} \sim Q} [-\log P(y|\mathbf{x}, \mathbf{w})]$. The latter term, which we refer to as average likelihood, penalises training errors. Therefore, the log marginal likelihood incorporates a similar trade-off as the PAC-Bayesian theorem, using the same complexity term up to log factors. If Q is just an approximation to the posterior, we see from (2.16) that the r.h.s. of (3.16) is a lower bound on $\log P(\mathbf{y})$. For a very broad model class, the average likelihood will be small, while for a small model class, the posterior Gibbs rule will mis-classify more points in the training sample, leading to a larger average likelihood. However, for a large model class the prior $P(\mathbf{w})$ is necessarily rather broad and the derivative $dQ(\mathbf{w})/dP(\mathbf{w})$ will be large in the region where the posterior $Q(\mathbf{w})$ is concentrated, leading to a large complexity term $n^{-1}D[Q \parallel P]$.

⁵This can be seen as strong instance of the so-called *stratification technique*, although we average over a continuum of possible $\tilde{\mathbf{w}}$ and do not use the union bound (see Section 2.2.1).

Put simply, the density ratio between discriminants we consider to be sensible after and before having seen the data, should increase with growth of the model class.

We would like to stress that the *number* of parameters of a model class is not a sensible measure of complexity. This is fairly obvious, since we can take any model class and create a new one by adding a large number of parameters which do not or only slightly influence the rules. Introducing the notion of “effective number of parameters” helps only if this number is defined unambiguously and can be computed feasibly. For example, the non-parametric methods we consider in Section 3.3 can be seen as having an infinite number of parameters, however these parameters are regularised by the prior, and the conditioning on a finite amount of data will only render a finite number of these parameters any significant influence on predictions. The complexity measure $D[Q \| P]$ behaves correctly in such situations, as the following argument suggests. Let $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$, and suppose that \mathbf{w}_2 has no influence on the rules, i.e. $P(y|\mathbf{x}, \mathbf{w}) = P(y|\mathbf{x}, \mathbf{w}_1)$. Then we have $dP(\mathbf{w}|\mathbf{y}) = dP(\mathbf{w}_1|\mathbf{y})dP(\mathbf{w}_2|\mathbf{w}_1)$ and $D[P(\mathbf{w}|\mathbf{y}) \| P(\mathbf{w})] = D[P(\mathbf{w}_1|\mathbf{y}) \| P(\mathbf{w}_1)]$, thus the complexity measure ignores \mathbf{w}_2 and its distribution.

We finally note that nowhere in the proof of Theorem 3.2 did we require a union bound (see Section 2.2.1), a distinctive advantage of the PAC-Bayesian method. In a nutshell, while traditional techniques often perform a sort of “sphere-covering” of a given hypothesis space, then employ some (often loose) covering number arguments and the union bound, the PAC-Bayesian theorem employs expectations instead, first over the prior P , then changing P for the posterior Q at the “cost” of $n^{-1}D[Q \| P]$. The slack comes from the fact that we use a linear lower bound to a convex function (see Section A.3).

3.2.4 The Case of General Bounded Loss

We stated and proved the PAC-Bayesian theorem above for the case of zero-one loss w.r.t. sets of size L . In this subsection, we provide a generalisation to general bounded loss functions in Theorem 3.3 below. The proof, which is given in Appendix B.3, uses a “water-filling” argument due to [117].

We adopt the notation of Section 3.2.2, but now assume that there is a bounded loss function $l(\tilde{\mathbf{w}}, (\mathbf{x}_*, y_*)) \in [0, 1]$ which quantifies the loss⁶ one occurs when using the rule $\tilde{\mathbf{w}}$ in order to predict the target corresponding to \mathbf{x}_* and the true target is y_* . For example, the zero-one loss for binary classification is given by $l(\tilde{\mathbf{w}}, (\mathbf{x}_*, y_*)) = \mathbb{I}_{\{y(\mathbf{x}_*|\tilde{\mathbf{w}}) \neq y_*\}}$. We use the notation $l(\tilde{\mathbf{w}}) = \mathbb{E}[l(\tilde{\mathbf{w}}, (\mathbf{x}_*, y_*))]$, where the expectation is over (\mathbf{x}_*, y_*) drawn from the data distribution, and

⁶The notation $l(\cdot)$ used here should not be confused with the notation $l \in \mathcal{L}$ used in subsection 3.2.2.

$\hat{l}(\tilde{\mathbf{w}}) = n^{-1} \sum_i l(\tilde{\mathbf{w}}, (\mathbf{x}_i, y_i))$. For fixed $\tilde{\mathbf{w}}$, $\hat{l}(\tilde{\mathbf{w}}) \rightarrow l(\tilde{\mathbf{w}})$ almost surely, and since l is bounded, the convergence rate is exponential in n . A typical large deviation inequality (see Section A.7) looks as follows. We have a nonnegative function ϕ on $[0, 1]^2$ such that for every fixed $\tilde{\mathbf{w}}$:

$$\begin{aligned} \Pr \left\{ \hat{l} \geq q \right\} &\leq e^{-n\phi(q,l)}, & q \geq l, \\ \Pr \left\{ \hat{l} \leq q \right\} &\leq e^{-n\phi(q,l)}, & q \leq l, \end{aligned} \quad (3.17)$$

where $l = l(\tilde{\mathbf{w}})$, $\hat{l} = \hat{l}(\tilde{\mathbf{w}})$. We require that $\phi(q, l)$ is nondecreasing in $|q - l|$ and furthermore convex in (q, l) , and $\phi(l, l) = 0$ for all $l \in [0, 1]$. For simplicity, we also require that $\phi(\cdot, l)$ is differentiable on $(0, l)$ and $(l, 1)$, for all $l \in (0, 1)$. We will give examples for possible inequalities in a moment. Choose some $\delta \in (0, 1)$.

Theorem 3.3 (PAC-Bayesian Theorem, Bounded Loss Functions) *For any data distribution over $\mathcal{X} \times \{-1, +1\}$, we have that the following bound holds, where the probability is over random i.i.d. samples $S = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$ of size n drawn from the data distribution:*

$$\Pr_S \left\{ \begin{array}{l} \phi \left(\mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\hat{l}(\tilde{\mathbf{w}})], \mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[l(\tilde{\mathbf{w}})] \right) > \frac{1}{n-1} \left(D[Q \parallel P] + \log \frac{2n+1}{\delta} \right) \\ \text{for some } Q \end{array} \right\} \leq \delta. \quad (3.18)$$

The proof is given in Appendix B.3. Note that the function $\phi(\mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\hat{l}(\tilde{\mathbf{w}})], \cdot)$ can be inverted just as easily as D_{Ber} above (see equation (3.1)) in order to obtain upper and lower bounds on $\mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[l(\tilde{\mathbf{w}})]$.

If we instantiate this theorem for zero-one loss using Chernoff's bound, we essentially obtain Theorem 3.1. In fact, Chernoff's bound holds just as well for general bounded loss and has the form of (3.17) with $\phi(q, l) = D_{\text{Ber}}[q \parallel l]$ (Theorem A.3). The original formulation in [117] uses the Hoeffding bound (see Appendix A.7) which is significantly less tight than the Chernoff bound if the expected empirical error is far from $1/2$. Other more specialised large-deviation inequalities can be used instead, and even if they do not come in the form of (3.17), the proof can probably be adapted. However, note that the constraint on $\phi(q, l)$ to be convex in (q, l) is rather crucial for the last step of the proof. More specifically, we have to upper bound $\phi(\mathbb{E}_Q[\hat{l}(\tilde{\mathbf{w}})], \mathbb{E}_Q[l(\tilde{\mathbf{w}})])$ in terms of the bound on $\mathbb{E}_Q[\phi(\hat{l}(\tilde{\mathbf{w}}), l(\tilde{\mathbf{w}}))]$, which is trivial if ϕ is convex.

3.2.5 An Extension to the Bayes Classifier

In practice, when comparing Gibbs and Bayes classifier for the same posterior distribution $Q(\mathbf{w})$ directly, it turns out that the Bayes variant often performs better than the Gibbs variant (the latter is hardly used in practice for this reason).

In this section, we will have a closer look on their relationship and suggest a simple extension of the PAC-Bayesian theorem for binary Bayes classifiers.

We have introduced Gibbs, Bayes and Bayes voting classifiers in Section 3.1.2. Recall that throughout this thesis we assume that the noise model is symmetric in the sense that it is a function of yu : $P(-y|u) = P(y|-u)$. For simplicity, we assume in this section that if a bias parameter b is used, it has already been added to u . In general, the Bayes, Bayes voting and predictive classifier are all different, but we will be interested exclusively in the case that the distribution of $u(\mathbf{x} | \mathbf{w})$ induced by Q is symmetric around its mean for every \mathbf{x} . In this case, all Bayes classifier variants agree. In fact, it is easy to see that for every nondecreasing $f(u)$ with $f(-u) + f(u) = \tau$, the classifiers $\text{sgn}(\mathbb{E}_Q[f(u(\mathbf{x}_* | \mathbf{w}))] - \tau/2)$ are identical. Note that if this assumption is violated, the concepts of Gibbs, Bayes and Bayes voting classifiers become questionable anyway and the predictive classifier should be used. Fortunately, all (approximate) Bayesian methods discussed in this thesis fulfil this assumption. Define the errors

$$\begin{aligned} e_{\text{Gibbs}}(\mathbf{x}_*, y_*) &= \mathbb{E}_Q[\mathbb{I}_{\{\text{sgn } u(\mathbf{x}_* | \mathbf{w}) \neq y_*\}}], \\ e_{\text{Bayes}}(\mathbf{x}_*, y_*) &= \mathbb{I}_{\{\text{sgn } \mathbb{E}_Q[u(\mathbf{x}_* | \mathbf{w})] \neq y_*\}}. \end{aligned}$$

Furthermore, for $A \in \{\text{Gibbs}, \text{Bayes}\}$, define $e_A = \mathbb{E}_{(\mathbf{x}_*, y_*)}[e_A(\mathbf{x}_*, y_*)]$ where the expectation is over the data distribution. Intuitively, the Bayes classifier should outperform the Gibbs variant if the trained model represents the data distribution well. Assume for now that the data distribution *is* identical to the model-generative one: for given \mathbf{x}_* , $y_* \sim P(y_* | u(\mathbf{x}_* | \mathbf{w}))$, $\mathbf{w} \sim Q$. Condition on \mathbf{x}_* and write $u = u(\mathbf{x}_* | \mathbf{w}) = \bar{u} + v$, where $\bar{u} = \mathbb{E}_Q[u]$ and v has an even density function. For simplicity, we write $u \sim Q$, $v \sim Q$ meaning the corresponding distributions induced by Q over \mathbf{w} (for fixed \mathbf{x}_*). Consider sampling $u_1, u_2 \sim Q$ independently, furthermore $y_2 \sim P(y_2 | u_2)$. Both Gibbs and Bayes classifier err if $\text{sgn } u_1 = \text{sgn } \bar{u}$, $y_2 \neq \text{sgn } u_1$, but if $\text{sgn } u_1 \neq \text{sgn } \bar{u}$, they differ depending on the relationship between y_2 and $\text{sgn } \bar{u}$. If $E = \{\text{sgn } u_1 \neq \text{sgn } \bar{u}\}$, then

$$e_{\text{Gibbs}} - e_{\text{Bayes}} = \mathbb{E}_{\mathbf{x}_*} \mathbb{E} \left[\Pr\{E\} (\Pr\{y_2 = \text{sgn } \bar{u}\} - \Pr\{y_2 \neq \text{sgn } \bar{u}\}) \mid \mathbf{x}_* \right].$$

$\Pr\{E\}$ is the probability of the tail $\{v \geq |\bar{u}|\}$ under Q . Note that we have used that the event $\{y_2 = \text{sgn } \bar{u}\}$ is independent of E , given \mathbf{x}_* . The conditional difference of the errors is positive if $\text{sgn } \bar{u} \neq 0$, showing that the Bayes classifier outperforms the Gibbs variant in this situation. On the other hand, it is easy to construct a ‘‘malicious’’ setting in which the Gibbs classifier does better than the Bayes variant (but see below), but recall from Section 3.1.1 that the relevance of such arguments may be limited in practice.

If nothing is known about the data distribution, we can relate e_{Bayes} and e_{Gibbs} in a coarser sense, by noting that if $e_{\text{Bayes}}(\mathbf{x}_*, y_*) = 1$, then $e_{\text{Gibbs}}(\mathbf{x}_*, y_*) \geq 1/2$,

thus $e_{\text{Bayes}} \leq 2 e_{\text{Gibbs}}$ (as remarked in [72], Lemma 5.3).⁷ Therefore, Theorem 3.1 applies to the Bayes classifiers as well. Although we obtained this extension without efforts, it is not really what we are ideally looking for. First, the bound on the Bayes classifier error is certainly over-pessimistic.⁸ Second, the generalisation error bound on the Bayes classifier is in terms of the expected empirical error of the *Gibbs* classifier: even if we prefer the Bayes classifier in practice, we have to evaluate its Gibbs variant in order to obtain performance guarantees. Third, the argument does not carry through to more than two classes. Very recently, Meir and Zhang [121] obtained a PAC-Bayesian margin bound for the Bayes voting classifier, combining a new inequality based on Rademacher complexities with the convex duality step. We discuss this result in Section 3.4.1. However, the Meir/Zhang result can be criticised on other grounds (see Section 3.4.1.1) and did not render non-trivial guarantees in our experiments (see Section 3.5.6), while the Gibbs theorem certainly did. In light of practical evidence, the aim is to provide a PAC-Bayesian theorem for Bayes-type classifiers which is at least as tight as the Gibbs theorem in practically relevant situations, and at least to our knowledge this remains an open problem.

3.2.6 Some Speculative Extensions

Here, at the end of Section 3.2 we take the liberty of collecting some more or less speculative thoughts which are not driven to a definite conclusion. In Section 3.2.6.1, we note that in principle the complexity measure $n^{-1}D[Q \parallel P]$ could be replaced by other such measures, as long as they are convex in Q for every P . In Section 3.2.6.2 we show that the essential slack in the PAC-Bayesian bound can be written down explicitly. Gaining knowledge about the behaviour of slack, especially about its degree of dependence on hyperparameters may be a key issue if PAC bounds are to be used for model selection in practice.

3.2.6.1 Generalisation to Other Complexity Measures

The nature of the proof given in Section 3.2.2 allows us to think about further generalisations. Recall that the key for transforming the uninteresting bound (3.9) into the statement of the PAC-Bayesian theorem is the characterisation (A.8) of the relative entropy $D[Q \parallel P]$ in terms of its convex dual (see Section A.3). Suppose that $d(P, Q)$ is any divergence measure between P and Q which is convex

⁷Even without any assumption on the distributions of $u(\mathbf{x}_* | \mathbf{w})$, this is true for *Bayes voting* and *Gibbs classifier*.

⁸One can construct situations to show that it is tight in principle, but recall from Section 3.1.1 that such “tightness” arguments may be misleading in practice.

in Q for all P . Then, there exists a convex dual $g(P, \lambda)$ such that

$$d(P, Q) = \max_{\lambda(\tilde{\mathbf{w}})} (\mathbb{E}_{\tilde{\mathbf{w}} \sim Q} [\lambda(\tilde{\mathbf{w}})] - g(P, \lambda)),$$

and $g(P, \lambda)$ is itself convex in λ for every P . For a general divergence measure $d(P, Q)$, the following program may be feasible to entertain. First, make sure that $d(P, Q)$ is convex in Q for every P . Second, determine its convex dual $g(P, \lambda)$. Third, prove a large deviation bound of the form

$$\Pr_S \left\{ e^{g(P, n \Delta(\tilde{\mathbf{w}}))} > \frac{K}{\delta} \right\} \leq \delta, \quad (3.19)$$

where K is polynomial in n . This would serve as equivalent for (3.9), and the generic part of the proof could be followed in order to obtain a PAC-Bayesian theorem which asserts that

$$D[\mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\hat{\mathbf{p}}(\tilde{\mathbf{w}})] \parallel \mathbb{E}_{\tilde{\mathbf{w}} \sim Q}[\mathbf{p}(\tilde{\mathbf{w}})]] \leq \frac{d(P, Q) + \log K - \log \delta}{n}$$

with probability at least $1 - \delta$ over draws of S . The feasibility of this approach depends on several factors. Of course, $d(P, Q)$ must be convex in Q and itself feasible to compute. Then, we have to determine the convex dual $g(P, \lambda)$ in closed form, or at least $g(P, n \Delta)$. Second (and probably most difficult) we have to prove the bound (3.19) for our prior at hand. Note that we are not necessarily constrained to use the particular divergence $\Delta(\tilde{\mathbf{w}})$ as defined in subsection 3.2.2: the latter was chosen for convenience w.r.t. proving (3.9). However, $\Delta(\tilde{\mathbf{w}})$ has to be convex in $(\hat{\mathbf{p}}, \mathbf{p})$ for every sample S . We have not yet pursued this program for any divergence $d(P, Q)$ other than the relative entropy.

3.2.6.2 The Slack Term in the PAC-Bayesian Theorem

Several authors have suggested to minimise PAC upper bounds in order to do model selection. In Section 2.2.4, we argue that this is theoretically justified only if one can prove that the *slack* in the bound, i.e. the difference between bound value and true generalisation error, is significantly less variable w.r.t. hyperparameters to be selected (over a range of interest) than the generalisation error itself. It is not even straightforward to formalise this requirement in the strict PAC sense, and proving it will most probably be much harder than the bound itself. Nevertheless, the phenomenon occurs in practice on non-trivial real world examples (see Section 3.5.5), and it would be very valuable to obtain some analytical results in order to understand it at least on toy models.

An interesting consequence of the simple proof of the PAC-Bayesian Theorem 3.1 given in Section 3.2.2 is that we can essentially write down the slack analytically. For simplicity, we deal with deterministic rules only, i.e. $\tilde{\mathbf{w}} = \mathbf{w}$.

Inspecting the proof, we see that there are three sources of slack: the initial arguments leading to (3.8), the plugging in of $n\Delta(\mathbf{w})$ for $\lambda(\mathbf{w})$ in (A.8) and the final application of Jensen's inequality in (3.15). The bound in (3.8) leads to a typically minor contribution to the PAC-Bayesian gap bound, and the final application of Jensen's inequality should be fairly tight if the posterior Q is rather concentrated,⁹ leaving us with the typically dominating slack coming from the "wrong" choice for $\lambda(\mathbf{w})$ in (A.8). From (3.14) we see that this slack in the right hand side $\varepsilon(\delta, S, P, Q)$ in (3.5) is $D[Q \parallel P_G]$, where P_G is the Gibbs measure defined in (3.13). Suppose that $Q(\mathbf{w})$ has the form

$$dQ(\mathbf{w}) = Z^{-1} e^{\sum_{i=1}^n \phi_i(\mathbf{w})} dP(\mathbf{w}), \quad Z = \mathbf{E}_{\mathbf{w} \sim P} \left[e^{\sum_{i=1}^n \phi_i(\mathbf{w})} \right],$$

where $\phi_i(\mathbf{w}) = \phi(\mathbf{x}_i, y_i; \mathbf{w})$. For example, the true Bayesian posterior is obtained if $\phi(\mathbf{x}_i, y_i; \mathbf{w}) = \log P(y_i | \mathbf{x}_i, \mathbf{w})$. Then,

$$\begin{aligned} D[Q \parallel P_G] &= \mathbf{E}_{\mathbf{w} \sim Q} \left[\log \frac{dQ(\mathbf{w})}{dP_G(\mathbf{w})} \right] \\ &= \sum_{i=1}^n \mathbf{E}_{\mathbf{w} \sim Q} [\phi_i(\mathbf{w}) - \Delta(\mathbf{w})] - \log \frac{\mathbf{E}_{\mathbf{w} \sim P} [e^{\sum_i \phi_i(\mathbf{w})}]}{\mathbf{E}_{\mathbf{w} \sim P} [e^{n\Delta(\mathbf{w})}]} \end{aligned}$$

We can also write

$$D[Q \parallel P_G] = -n \mathbf{E}_{\mathbf{w} \sim Q} [\Delta(\mathbf{w})] + \log \mathbf{E}_{\mathbf{w} \sim P} [e^{n\Delta(\mathbf{w})}] - H[Q(\mathbf{w})].$$

Admittedly, these expressions are not very useful per se, except for showing that the slack will be small if $\Delta(\mathbf{w})$ is close to most of the $\phi_i(\mathbf{w})$.

3.3 Application to Gaussian Process Classification

In this section, we apply the PAC-Bayesian Theorem 3.1 to a wide class of Gaussian process models for binary classification. The class is defined in Section 2.1.3, and in Section 3.3.1 we show how the bound terms can be computed for any member. In Sections 3.3.2, 3.3.3 and 3.3.4, we give specific examples for some well-known GP approximations (see Section 2.1.3).

3.3.1 PAC-Bayesian Theorem for GP Classification

In this section, we specialise the PAC-Bayesian Theorem 3.1 to Gaussian process models for binary classification, incorporating a wide class of approximate inference methods. The GP binary classification model has been introduced in

⁹This could be tested using random sampling for a particular architecture.

Section 2.1.2 (both the logit and probit noise model discussed there satisfy the symmetry condition required here). Recall that $\mathbf{K} \in \mathbb{R}^{n,n}$ denotes the covariance matrix evaluated over the training input points $\{\mathbf{x}_i\}$, and $\mathbf{u} = (u_i)_i \in \mathbb{R}^n$ are the latent outputs at these points. This non-parametric model can be seen as special case of the scenario of Section 3.1.2 with $\mathbf{w} \equiv u(\cdot)$, i.e. the “weights” are the complete latent process, and $u(\mathbf{x}|\mathbf{w}) \equiv u(\mathbf{x})$. Alternatively, we could develop the process $u(\mathbf{x})$ in an eigensystem of the covariance kernel K and parameterise it in terms of a countable number of weights (the “weight space view” of Section 2.1.1), however the presentation in the “process view” turns out to be much simpler.

The general class of approximation methods we are interested in here is defined in Section 2.1.3. The posterior $P(\mathbf{u}|S)$ is approximated by a Gaussian $Q(\mathbf{u}|S)$ of the general form (2.12). For most schemes, the covariance matrix \mathbf{A} of $Q(\mathbf{u}|S)$ is further restricted to the form (2.13), thus the approximation is defined in terms of \mathbf{K} and further $O(d)$ parameters, $d \leq n$. Then, the approximate predictive distribution is Gaussian with mean and variance given in (2.14). In order to apply the PAC-Bayesian theorem to this case, we only have to show how to compute the terms defining the gap bound value: the expected empirical error and the relative entropy $D[Q \| P]$. The former is just the empirical average over

$$e_{\text{Gibbs}}(\mathbf{x}_*, y_*) = \Pr_{u_* \sim Q(u_* | \mathbf{x}_*, S)} \{ \text{sgn}(u_* + b) \neq y_* \} = \Phi \left(\frac{-y_* (\mu(\mathbf{x}_*) + b)}{\sigma(\mathbf{x}_*)} \right), \quad (3.20)$$

where $\Phi(\cdot)$ denotes the c.d.f. of $N(0, 1)$. The relative entropy $D[Q \| P]$ has been determined in (2.11). Using (A.17), we obtain

$$D[Q \| P] = \frac{1}{2} \left(\log |\mathbf{A}^{-1} \mathbf{K}| + \text{tr} (\mathbf{A}^{-1} \mathbf{K})^{-1} + \boldsymbol{\xi}^T \mathbf{K}_I \boldsymbol{\xi} - n \right). \quad (3.21)$$

If \mathbf{A} is of the special form (2.13), this simplifies to

$$D[Q \| P] = \frac{1}{2} \left(\log |\mathbf{B}| + \text{tr} \mathbf{B}^{-1} + \boldsymbol{\xi}^T \mathbf{K}_I \boldsymbol{\xi} - d \right), \quad (3.22)$$

where \mathbf{B} is defined in (2.14). These formulae depend on the generic parameters $\boldsymbol{\xi}$ and \mathbf{A} (or I and \mathbf{D}) of the posterior approximation $Q(\mathbf{u}|S)$. In Sections 3.3.2 and 3.3.3, we show how to compute the relative entropy for a range of concrete GPC approximations. A simple way to compute (3.22) is to use the Cholesky decomposition $\mathbf{B} = \mathbf{L}\mathbf{L}^T$ (see Appendix A.2.2). Then, $\log |\mathbf{B}| = 2 \log |\text{diag}^2 \mathbf{L}|$, and $\text{tr} \mathbf{B}^{-1}$ can be computed from \mathbf{L} using the same number of operations as required for $\mathbf{B} \rightarrow \mathbf{L}$.

We can now simply plug in (3.21) or (3.22) and (3.20) into the terms of Theorem 3.1 in order to obtain a PAC-Bayesian theorem for GP binary Gibbs classifiers. It is important to note that for this theorem to be valid, the prior

P and the model have to be fixed *a priori*, i.e. free hyperparameters of the covariance function K or the noise model $P(y|u)$ have to be chosen *independently* of the training sample S . For example, the widely used practice of choosing such parameters by maximising an approximation to the log marginal likelihood $\log P(S)$ (see Sections 2.1.3 and A.6.2) is *not* compatible with using the theorem for a statistical test. It is easy to modify Theorem 3.1 to allow for model selection amongst a finite set (size polynomial in n) of hyperparameter candidates, by a straightforward application of the union bound (see Section 2.2.1). This introduces a $O(\log n)$ term in the numerator of $\varepsilon(\delta, n, P, Q)$ in (3.3). However, choosing the hyperparameters by continuous optimisation is not admissible.

Note also that Theorem 3.2 in principle applies to multi-class GP methods (see Section 2.1.2). As mentioned there, the implementation of such methods is, however, quite involved, and for most methods, additional approximations are required to avoid a scaling quadratic in the number of classes. In terms of Theorem 3.2, the computation of $\hat{p}(S, Q)$ may not be analytically tractable, and special approximations may have to be considered in order not to compromise the bound statement. This program is subject to future work.

3.3.2 Laplace Gaussian Process Classification

Let us specialise the generic framework of the previous section to Laplace GP binary classification [212], discussed briefly in Section 2.1.3. We end up with a Gaussian posterior approximation with parameters $\boldsymbol{\xi}$ and covariance $\mathbf{A} = (\mathbf{K}^{-1} + \mathbf{D})^{-1}$, where $\boldsymbol{\xi}$, \mathbf{D} depend on the posterior mode $\hat{\mathbf{u}}$ via (2.15). Note that in order to evaluate the Gibbs classifier, the predictive variance $\sigma^2(\mathbf{x}_*)$ has to be computed, while the (approximate) Bayes classifier is $\text{sgn}(\mu(\mathbf{x}_*) + b)$, independent of $\sigma^2(\mathbf{x}_*)$ (see Equation 2.14).¹⁰ The downside of this is that the Gibbs classifier requires $O(n^2)$ for each evaluation, while the Bayes classifier is $O(n)$ if an uncertainty estimate is not required. We show in Appendix B.4 how to evaluate the Gibbs classifier more efficiently using sparse approximations together with rejection sampling techniques, resulting in $O(n)$ (average case) per prediction.

Experimental results are given in Section 3.5. We can gain some insight into the gap bound value by analysing the relative entropy term $D[Q \| P]$ (see Equation 3.22) in Theorem 3.1, as applied to Laplace GPC. In normal situations (i.e. δ not extremely small), the expression $\varepsilon(\delta, n, P, Q)$ in (3.3) is dominated by this term.

We assume that $P(y|u)$ is logit noise. First, use (2.15) to see that $\boldsymbol{\xi}^T \mathbf{K} \boldsymbol{\xi} = \hat{\mathbf{u}}^T \boldsymbol{\xi} = \sum_i y_i \hat{u}_i \sigma(-y_i \hat{u}_i) = \sum_i f(y_i \hat{u}_i)$, where $f(x) = x \sigma(-x)$ and $y_i \hat{u}_i$ is the

¹⁰Recall from Section 3.1.2 that since $Q(u_* | \mathbf{x}_*, S)$ is symmetric around $\mu(\mathbf{x}_*)$, Bayes, Bayes voting and predictive classifiers predict the same target. The independence of these classifiers of $\sigma^2(\mathbf{x}_*)$ could be interpreted as weakness of the Gaussian approximation, since the true posterior for u_* can be skew. Error bars for the classifiers *do* of course depend on the predictive variance.

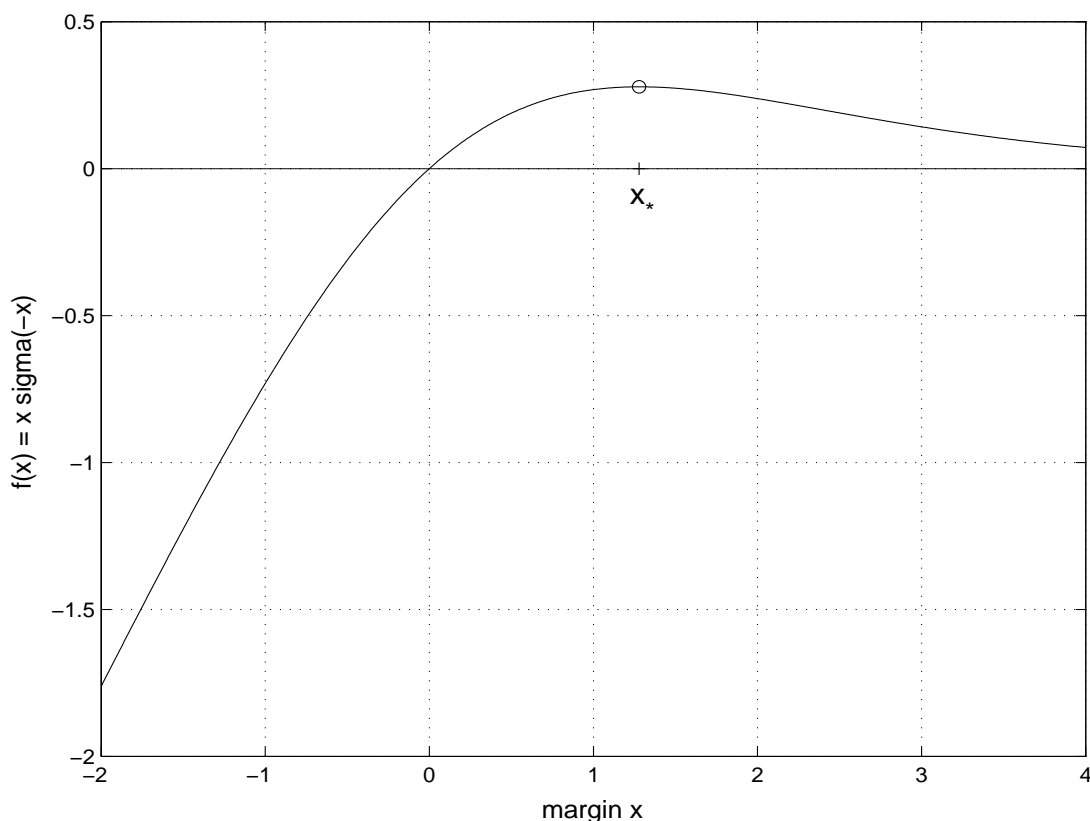


Figure 3.1: Relation between margin and gap bound part

margin¹¹ at example (\mathbf{x}_i, y_i) . $f(x)$ is plotted in Figure 3.1. It is maximal at $x_* \approx 1.28$, converges to 0 exponentially quickly for $x \rightarrow \infty$ and behaves like $x \mapsto x$ for $x \rightarrow -\infty$. Thus, at least w.r.t. the third term in (3.22), classification mistakes (i.e. $y_i \hat{y}_i < 0$) render a negative contribution to the gap bound value. This is what we expect for a Bayesian architecture. Namely, the method choses a *simple* solution, at the expense of making this mistake, yet with the goal to prevent disastrous over-fitting. In our bound, we are penalised by a higher expected empirical error, but we should be rewarded by a smaller gap bound term. Now to the remaining terms in (3.22). The matrix $\mathbf{B} = \mathbf{I} + \mathbf{D}^{1/2} \mathbf{K} \mathbf{D}^{1/2}$ is positive definite, and all its eigenvalues are ≥ 1 . Furthermore, by taking any unit vector \mathbf{z} and using the Fisher-Courant min-max characterisation of the eigenvalue spectrum of Hermitian matrices (e.g., [78], Sect. 4.2), we have $\mathbf{z}^T \mathbf{B} \mathbf{z} = 1 + (\mathbf{D}^{1/2} \mathbf{z})^T \mathbf{K} (\mathbf{D}^{1/2} \mathbf{z}) \leq 1 + \lambda_{\max} \mathbf{z}^T \mathbf{D} \mathbf{z} < 1 + \lambda_{\max}/4$, where λ_{\max} is the largest eigenvalue of \mathbf{K} . Here, (2.15) implies that the coefficients of $\text{diag } \mathbf{D}$ are all $< 1/4$. Thus, all eigenvalues

¹¹The margin is a learning-theoretical concept frequently used in the context of PAC bounds for averaging (Bayes) classifiers (see for example Section 3.4.1).

of \mathbf{B} lie in $(1, 1 + \lambda_{\max}/4)$. By analysing $(1/2) \log |\mathbf{B}| + (1/2) \text{tr} \mathbf{B}^{-1}$ for general¹² $\mathbf{B} \succeq \mathbf{I}$, using a spectral decomposition of \mathbf{B} , we see that this term must lie between $n/2$ and $(n/2)(\log(1 + \lambda_{\max}/4) + (1 + \lambda_{\max}/4)^{-1})$, although these bounds are not very tight.

3.3.3 Sparse Greedy Gaussian Process Classification

In practice, the applicability of Gaussian process or other kernel methods is often severely restricted by their unfortunate scaling: $O(n^3)$ time for first-level inference, $O(n^2)$ space. Sparse approximations to GP models can improve vastly on these figures and are therefore of considerable practical importance. In Chapter 4, we discuss sparse approximations in detail and present a range of different schemes for approximate inference. All of these lie within the class described in Section 2.1.3, so that Theorem 3.1 applies to their Gibbs classifier versions. In this section, we focus on the simple IVM scheme introduced in Section 4.4.1, the most efficient of the algorithms discussed in this thesis.

The parametric representation for IVM is described in Section 4.4.1 and Appendix C.3.1. In terms of the placeholders of Section 2.1.3 we have $\mathbf{D} = \mathbf{\Pi}_I$, where I is the active set, furthermore $\boldsymbol{\xi} = \mathbf{I}_{\cdot, I} \mathbf{\Pi}_I^{1/2} \mathbf{L}^{-T} \boldsymbol{\beta}$. The relative entropy term is given by

$$D[Q \parallel P] = \frac{1}{2} \left(\log |\mathbf{B}| + \text{tr} \mathbf{B}^{-1} + \|\boldsymbol{\beta}\|^2 - \|\mathbf{L}^{-T} \boldsymbol{\beta}\|^2 - d \right).$$

The predictive distribution $Q(u_* | \mathbf{x}_*, S)$ is derived in Section 4.4.1. An evaluation of the Gibbs classifier costs $O(d^2)$ (one back-substitution with \mathbf{L}), while the computation of $D[Q \parallel P]$ costs $O(d^3)$. The expected empirical error can be computed in $O(n d^2)$.

As discussed in Section 4.3, the non-sparse equivalent of the IVM (i.e. $I = \{1, \dots, n\}$) is the cavity TAP method of [139], and Theorem 3.1 can be applied to this approximation using the formulae given here. Also note that although the IVM is a compression scheme (in the sense defined in Appendix B.6), the PAC-Bayesian theorem is *not* a compression bound: it holds for sparse non-compressing methods (such as PLV, see Section 4.4.2) just as well. We will see in Section 3.5.4 that the PAC-Bayesian theorem can be significantly tighter in practice than a standard PAC compression bound when applied to the same compression scheme.

¹² $\mathbf{B} \succeq \mathbf{I}$ means that $\mathbf{B} - \mathbf{I}$ is positive semidefinite. See [24] for details about such generalised inequalities.

3.3.4 Minimum Relative Entropy Discrimination

The MRED framework as probabilistic interpretation of large margin (discrimination) methods such as SVM has been introduced in [83] and is discussed in Section 2.1.6. The formulation allows a direct application of the PAC-Bayesian theorem 3.1 to Gibbs version of the corresponding classifiers.

Recall that the prior process P and the MRED “posterior” process Q have the same covariance, thus

$$D[Q \parallel P] = \frac{1}{2} \boldsymbol{\xi}^T \mathbf{K} \boldsymbol{\xi} = \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{Y} \mathbf{K} \mathbf{Y} \boldsymbol{\lambda}.$$

Therefore, the application of Theorem 3.1 to MRED is straightforward. However, note that the Gibbs variant of the MRED classifier is typically not a very useful method in practice, due to the failure of MRED (and large margin methods in general) to provide non-trivial estimates of predictive variance. The predictive variance of the latent variable u_* is the same as the corresponding prior variance which can be large, thus the Gibbs variant is likely to perform much worse than the Bayes (averaging) one for points close to a decision boundary even if they lie close to training points.

3.4 Related Work

In this section, we mention some closely related work. Our focus is on a result very recently proved in [121], giving a PAC-Bayesian theorem for Bayes voting classifiers, as opposed to Theorem 3.1 which holds for Gibbs classifiers. In Section 3.4.1.2, we show a possible route towards extending this result to the case of GP regression.

The literature on PAC bounds for kernel methods is large and will not be reviewed here (the reader may consult [72, 162, 199]). We provide a brief introduction to PAC bounds in Section 2.2. Shawe-Taylor and Williamson [177] present a PAC analysis of a Bayesian estimator. The notion of PAC-Bayesian theorems has been developed by McAllester [116, 115, 117] who also gave a proof of a slightly less tight version of Theorem 3.1. The theorems in [116] can be seen as special case for a restricted class of “cut-off posteriors” (where the log likelihood is an indicator function). McAllester’s theorems have been used in a number of later papers. Herbrich and Graepel [73] use the theorems in [116] to obtain a margin bound on SVM. Seeger, Langford, and Megiddo [173] combine the “sampling” approach from [160] with the PAC-Bayesian theorem to obtain a Bayes classifier version. Langford and Caruana [95] apply McAllester’s theorem to multi-layer perceptrons. Langford and Shawe-Taylor [94] provide another extension of Theorem 3.1 to Bayes classifiers, however with the same drawbacks as the one in Section 3.2.5. The interesting feature of this work is that it shows how

to apply the PAC-Bayesian theorem to averaging (Bayes) classifiers which do not provide estimates of predictive variances. A similar idea has been proposed in [73], however with an unfortunate dependence on the dimensionality of the weight space. There is a considerable literature on PAC bounds for combined (mixture) classifiers and multi-layered classifiers. Recent results are given by Koltchinskii and Panchenko [91] where other references can be found, see also [160].

3.4.1 The Theorem of Meir and Zhang

The PAC-Bayesian Theorem 3.1 holds for Gibbs classifiers only, while in practice Bayes or Bayes voting classifiers are used much more frequently, due to their typically better performance. Recently, Meir and Zhang [121] presented a PAC-Bayesian margin bound which applies to Bayes voting classifiers (recall that for the approximations we are interested in, Bayes and Bayes voting classifiers are identical, see Section 3.1.2). They obtain their result by combining a recent bound proved in [91] with a convex duality step identical to the second part of our proof. In this section, we present their result and compare it with the PAC-Bayesian Theorem 3.1. In fact, we derive a slightly different theorem which is closer to the relevant Theorem 2 in [91].

Meir and Zhang [121] consider functions $f(\mathbf{x}|Q) = \mathbb{E}_{\mathbf{w} \sim Q}[y(\mathbf{x}|\mathbf{w})]$, $y(\mathbf{x}|\mathbf{w}) = \text{sgn}(u(\mathbf{x}|\mathbf{w}) + b)$. Allowing for the choice of a prior P , they define

$$\mathcal{Q}_A = \{Q \mid D[Q \parallel P] \leq A\}, \quad \mathcal{F}_A = \{f(\mathbf{x}|Q) \mid Q \in \mathcal{Q}_A\}.$$

Note that Meir and Zhang claim that their theorem holds for arbitrary classes \mathcal{F}_A , while based on the theorems in [91], we were only able to reproduce their result under the additional condition¹³ that all \mathcal{F}_A are closed under multiplication with -1 . In other words, we require that for every $A > 0$ and every $f \in \mathcal{F}_A$: $-f \in \mathcal{F}_A$. Similar to structural risk minimisation bounds [199], they use a nested hierarchy $\{\mathcal{F}_{A_j}\}$ defined *a priori* via an unbounded sequence $A_1 < A_2 < \dots$ together with a distribution $(p_j)_j$ over \mathbb{N} . Finally, let $\phi(x)$ be the function which is equal to $1 - x$ in $[0, 1]$, constant 1 for $x \leq 0$ and constant 0 for $x \geq 1$. Now, choose a $\delta \in (0, 1)$ and a $c > 1$.

Theorem 3.4 ([121]) *For any data distribution over $\mathcal{X} \times \{-1, +1\}$ we have that the following bound holds, where the probability is over random i.i.d. samples*

¹³The problem will be removed in a longer version of their paper (Tong Zhang, personal communication).

$S = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$ of size n drawn from the data distribution:

$$\Pr_S \left\{ \Pr_{(\mathbf{x}_*, y_*)} \{y_* f(\mathbf{x}_* | Q) \leq 0\} > \inf_{\kappa \in (0,1]} B(\kappa, S, Q) + \sqrt{\frac{\log(2/(p_j(Q)\delta))}{2n}} - \frac{\log \log c}{n} \text{ for some } Q \right\} \leq \delta.$$

Here,

$$B(\kappa, S, Q) = \frac{1}{n} \sum_{i=1}^n \phi(y_i f(\mathbf{x}_i | Q) / \kappa) + \frac{4c}{\kappa} \sqrt{\frac{2A_j(Q)}{n}} + \frac{\log \log(2/\kappa)}{n}$$

and $j(Q) = \min\{j \mid D[Q \parallel P] \leq A_j\}$.

The proof of the theorem is given in Appendix B.5. Note that the search for a minimiser κ is equivalent to an optimisation w.r.t. margin loss functions $\phi(\cdot/\kappa)$. Both $c > 1$ and the hierarchy (A_j, p_j) have to be chosen *a priori*. As long as they remain within sensible limits, the effect of their precise choice is insignificant: the bound value is typically dominated by the first two terms in $B(\kappa, S, Q)$. We may choose $c = 1.01$, $A_j = j n (\Delta A)$, $p_j = 1/(j(j+1))$, where $\Delta A > 0$ is a grid size.

3.4.1.1 Comparing the PAC-Bayesian Theorems

Theorem 3.4 is quite different in form from Theorem 3.1 and the corresponding statement for the Bayes classifier mentioned in Section 3.2.5. A direct analytical comparison is difficult, because we would expect both theorems to show their merits only in “lucky” cases (which frequently occur in practice). In this section, we give some comparative arguments, pointing out a serious lack in tightness in the result of Meir and Zhang. An empirical comparison is presented in Section 3.5.6.

We are interested in the GP binary classification situation. Let $r_i = (\mu(\mathbf{x}_i) + b)/\sigma(\mathbf{x}_i)$. From (3.20) we know that

$$\hat{e}_{\text{Gibbs}} = \frac{1}{n} \sum_{i=1}^n \Phi(-y_i r_i).$$

It is also easy to see that $f(\mathbf{x}_i | Q) = 2\Phi(r_i) - 1$, so that

$$\frac{1}{n} \sum_{i=1}^n \phi(\kappa^{-1} y_i f(\mathbf{x}_i | Q)) = \frac{1}{n} \sum_{i=1}^n \phi(\kappa^{-1} (2\Phi(y_i r_i) - 1)).$$

The Bayes classifier makes a mistake whenever $y_i r_i \leq 0$, in which case we have $\phi(\kappa^{-1} (2\Phi(y_i r_i) - 1)) = 1$ independent of κ , while the contribution to the Gibbs

empirical error can still be close to $1/2$ if $|r_i|$ is small. The reason why the Bayes variant often outperforms the Gibbs variant can only lie with patterns for which $|r_i|$ is fairly small, because Gibbs will with high probability make the same prediction as Bayes if $|r_i| \gg 0$. In these situations, among the patterns with small $|r_i|$ there are significantly more “on the right side”, i.e. $y_i r_i > 0$. The contribution to the margin loss is smaller than the one to the Gibbs error once $\Phi(y_i r_i) > 1/(2 - \kappa)$, and in this region the margin loss drops to zero quickly (slope κ^{-1}) while the Gibbs contribution is ≈ 0 only for $|r_i| > 2$ (see Figure 3.2).

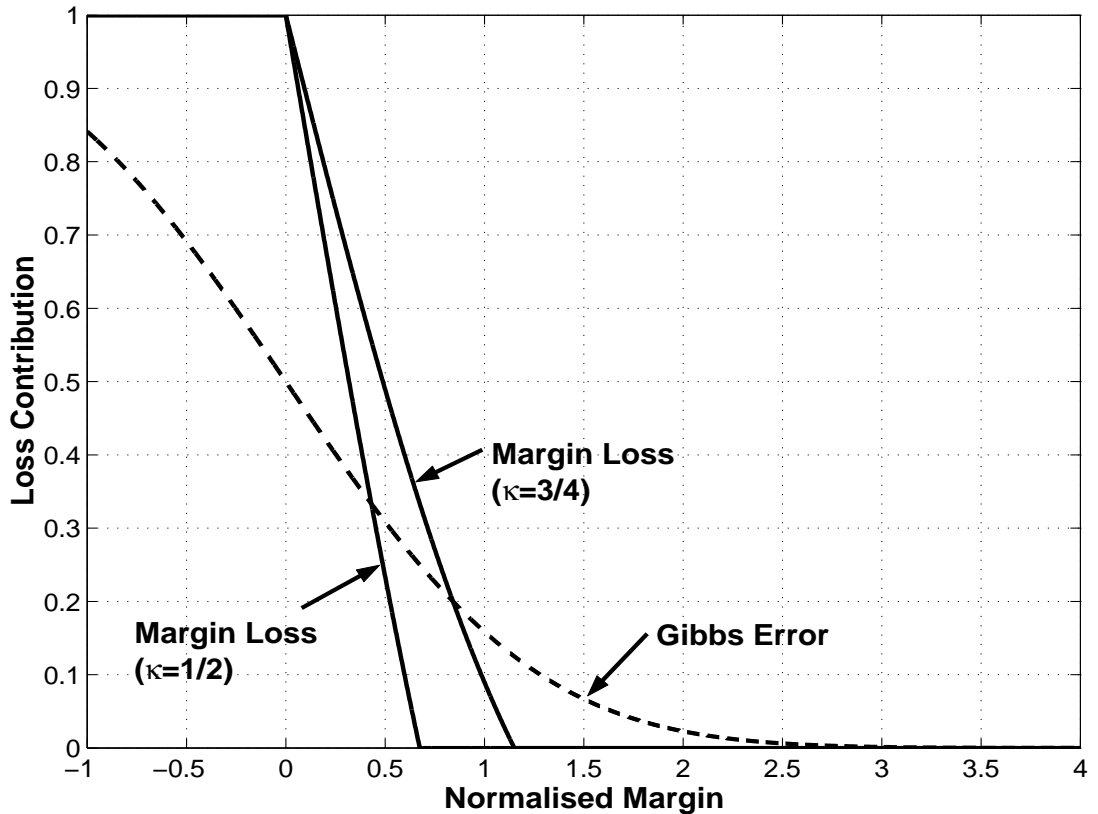


Figure 3.2: Contribution of single pattern to expected Gibbs error and Bayes margin loss respectively. The horizontal unit is the “normalised margin” $y_i r_i = y_i(\mu(\mathbf{x}_i) + b)/\sigma(\mathbf{x}_i)$.

The advantage of the margin loss over the expected Gibbs error grows with smaller κ , but the other dominant term in the bound of Theorem 3.4 scales as $1/\kappa$. This second term poses a serious problem in the Bayes bound, since it scales as $(D[Q \| P]/n)^{1/2}$ instead of $D[Q \| P]/n$ in the Gibbs bound. This becomes an issue w.r.t. tightness if the empirical Bayes error and the empirical margin loss is much smaller than $1/2$, as will usually be the case in practice. It is interesting that it is exactly this case that the authors put forward in [121] to contrast their

result with McAllester's. They point out rightly that one should rather focus PAC studies on the Bayes than on the Gibbs classifier, given that the former usually has much lower error rates in practice. However, their bound fails to exploit properly this case of low error rates far from $1/2$. In classical VC bounds which deal with zero-one loss, this problem is elegantly circumvented by bounding relative deviations such as $(p - \hat{p})/\sqrt{\hat{p}}$ (see [198, 5]). A bound $(p - \hat{p})/\sqrt{\hat{p}} \leq \varepsilon$ implies

$$p \leq \hat{p} + \frac{1}{2}\varepsilon^2 \left(1 + \sqrt{1 + 4\hat{p}/\varepsilon^2}\right),$$

so that the bound on the gap $p - \hat{p}$ is $O(\varepsilon)$ in the worst case $\hat{p} = 1/2$, but can be much smaller if \hat{p} is small. It would be interesting to find out whether this technique carries over to the PAC-Bayesian theorem.¹⁴ In fact, recently exponential concentration inequalities have been provided which exploit a small variance of the unknown process [21, 20], these could possibly serve as a substitute of the bounded difference inequality used in the present proof. Furthermore, the critical term drops below 1 only if $32D[Q \| P] < n$, which may not happen at all in practice (see Section 3.5.6).

3.4.1.2 Towards a PAC-Bayesian Theorem for Regression

How could we obtain a PAC-Bayesian theorem for a regression model? We are not aware of the existence of a theorem comparable in tightness to Theorem 3.1 or Theorem 3.4 for the regression setting. Note that while Gibbs versions of Bayes-like classifiers often show acceptable performance in practice, a Gibbs variant of a regression estimator would not be sensible, since the estimate is required to be smooth. In this section, we present some thoughts towards a PAC-Bayesian theorem for regression, however without coming to a definite conclusion. Readers not interested in speculative material should jump to the next section. The details are given in Appendix B.5.1.

Recall the GP regression model from Section 2.1.2. The latent process $u(\mathbf{x}|\mathbf{w})$, *a priori* Gaussian with kernel K , is obscured by independent noise $P(y|u)$ before observation. We will use a non-negative loss function $\phi(\cdot)$ which is Lipschitz with maximum slope κ , and $\phi(0) = 0$. Loss is quantified as $\phi(y - f(\mathbf{x}))$, for example the empirical risk is

$$\frac{1}{n} \sum_{i=1}^n \phi(y_i - \mathbb{E}_{\mathbf{w} \sim Q}[u(\mathbf{x}_i|\mathbf{w})]). \quad (3.23)$$

Note that if ϕ is also convex (Huber or Laplace loss are examples of ϕ which are Lipschitz and convex, see Section A.6.1), we could obtain a bound on the risk

¹⁴The proof of the zero-one loss situation does not carry over straightforwardly. Symmetrisation by a ghost sample and randomisation can be done, but it is not clear (to us) how to get rid of the margin loss function (in the proof of Theorem 3.4 this is done using a contraction principle, see Section B.5).

from a bound on the risk of the not very useful, but theoretically possible Gibbs regression estimate:

$$\mathbb{E}_{(\mathbf{x}_*, y_*)} [\phi(y_* - \mathbb{E}_{\mathbf{w} \sim Q}[u(\mathbf{x}_* | \mathbf{w})])] \leq \mathbb{E}_{(\mathbf{x}_*, y_*)} [\phi(y_* - u(\mathbf{x}_* | \mathbf{w}))].$$

However, such a bound would most probably be in terms of the empirical risk of the Gibbs estimate, which can be significantly larger than the empirical risk of the Bayes estimate.

We can try to move along the lines of Theorem 3.4. The first steps in the proof of Theorem 1 in [91] make use of Hoeffding's bounded differences inequality (see Section 2.2.2) which is not directly applicable to the regression setting with unbounded loss. Therefore, in order to prove a theorem in the regression setting, a different concentration argument would have to be used. To this end, it is probably necessary to make some sort of assumptions (such as bounded variance) on the distribution of the targets.

However, the remaining part of the proof, namely the bounding of the term $\mathbb{E} \|P_n - P\|_{\mathcal{G}_\phi}$ can be done, as is shown in Appendix B.5.1. Here, \mathcal{G}_ϕ consists of the functions $(\mathbf{x}, y) \mapsto \phi(y - \mathbb{E}_{\mathbf{w} \sim Q} u(\mathbf{x} | \mathbf{w}))$, $Q \in \mathcal{Q}_A$:

$$\mathbb{E} \|P_n - P\|_{\mathcal{G}_\phi} \leq \frac{4}{\kappa \sqrt{n}} \left(\mathbb{E}_S \left[\sqrt{2 A(\text{tr } \mathbf{K}/n)} \right] + \sqrt{\mathbb{E}[y^2]} \right). \quad (3.24)$$

Here, $\mathbb{E}[y^2]$ is an expectation over the data distribution. Furthermore, $n^{-1} \text{tr } \mathbf{K}$ should concentrate under mild conditions on the data distribution and/or the kernel function. For example, if K is a stationary kernel (see Section 2.1.1), then $K(\mathbf{x}, \mathbf{x}) = C$ for every \mathbf{x} , and $\text{tr } \mathbf{K}/n = C$, the prior process variance. If we could show concentration of $\|P_n - P\|_{\mathcal{G}_\phi}$ under a boundedness condition on $\mathbb{E}[y^2]$, we would end up with a PAC-Bayesian theorem for the GP regression model.

3.5 Experiments

In this section, we present experiments testing instantiations of the PAC-Bayesian Theorem 3.1 for the Laplace GP Gibbs classifier in Section 3.5.2 and the sparse greedy GP (IVM) Gibbs classifier in Section 3.5.3 (these special cases of the bound are described in Sections 3.3.2 and 3.3.3). The results indicate that the bounds are very tight even for training samples of moderate sizes. In Section 3.5.4, we compare our bound to a state-of-the-art PAC compression bound for the IVM Bayes classifier, and to the same compression bound for the soft-margin support vector classifier in Section 3.5.4.1. In Section 3.5.5 we try to evaluate the model selection qualities of the PAC-Bayesian bound, again applied to the IVM Gibbs classifier. Finally, in Section 3.5.6 we strengthen some of the results by repeating experiments on a different setup and present comparisons with the bound of Meir and Zhang (see Section 3.4.1).

3.5.1 The Setup MNIST2/3

Let us describe the experimental design MNIST2/3 for most of the sections to come. This design applies in all sections but the last one (3.5.6). A real-world binary classification task was created on the basis of the well-known MNIST handwritten digits database¹⁵ as follows. MNIST comes with a training set of 60000 and a test set of 10000 handwritten digits, represented as 28-by-28-pixel bitmaps, the pixel intensities are quantised to 8 bit values. First, the input dimensionality was reduced by cutting away a 2-pixel margin, then averaging intensities over 3-by-3-pixel blocks, resulting in 8-by-8-pixel bitmaps. We concentrated on the binary subtask of discriminating twos against threes, for which a training pool of 12089 cases and a test set of $l = 1000$ cases were created.

The Gaussian process prior in our model was parameterised by a Gaussian (RBF) kernel with variance parameter $C > 0$ and an inverse squared length scale $w > 0$ (2.27). Since all the tasks we considered are balanced, we did not employ a bias parameter in the noise model.

The experimental setup was as follows. An experiment consisted of ten independent iterations. During each iteration, three datasets were sampled independently and without replacement from the training pool: a model selection (MS) training set of size n_{MS} , a MS validation set of size l_{MS} and a task training sample S of size n . Note that the latter set is sampled independently from the model selection sets, ensuring that the prior P in Theorem 3.1 is independent of the task training sample. Then, model selection was performed over a list of candidates for (w, C) , where a classifier was trained on the MS training set and evaluated on the MS validation set (the MS score was the expected empirical error of the Gibbs classifier on the MS validation set). The winner was then trained on the task training set S and evaluated on the test set. Alongside, the upper bound value given by Theorem 3.1 was evaluated. The confidence parameter δ was fixed to 0.01.¹⁶ We also quote total running times for some of the experiments. Where timing figures are given, these have been obtained on the same machine using the same implementation.

3.5.2 Experiments with Laplace GPC

The Laplace approximation framework for GP binary classification is described in Section 2.1.3, and the application of the PAC-Bayesian theorem to this method in Section 3.3.2. We used a logit noise model without a bias term, $P(y|u) = \sigma(yu)$, σ the logistic function. Note that both the computation of the relative entropy term and the expected empirical error require $O(n^3)$, and predictions

¹⁵Available online at <http://www.research.att.com/~yann/exdb/mnist/index.html>.

¹⁶Note that much smaller values of δ could have been used without altering the upper bound values significantly.

are $O(n^2)$ per pattern (we did not employ the sampling techniques from Appendix B.4). Note also that the complete kernel matrix \mathbf{K} has to be evaluated and stored. Our implementation requires only one buffer of size n^2 .

The specifications and results for the experiments of this section are listed in Table 3.1. For all these experiments, we chose model selection validation set size $l_{\text{MS}} = 1000$ (recall that the test set is fixed with size $l = 1000$). Experiments #1 to #5 have growing sample sizes $n = 500, 1000, 2000, 5000, 9000$, the corresponding MS training set sizes are $n_{\text{MS}} = 1000$ for experiments #2 to #5, and $n_{\text{MS}} = 500$ for experiment #1. Note that $n_{\text{MS}} < n$ in experiments #3 to #5 is chosen for computational feasibility, due to the considerable size of the candidate list for (C, w) .

#	n	n_{MS}	emp	gen	upper	gen-bayes	time
1	500	500	0.036 (± 0.0039)	0.0469 (± 0.0015)	0.182 (± 0.0057)	0.0339 (± 0.0023)	14
2	1000	1000	0.0273 (± 0.0023)	0.036 (± 0.001)	0.131 (± 0.0041)	0.0274 (± 0.0022)	67
3	2000	1000	0.0243 (± 0.0026)	0.028 (± 0.0013)	0.1091 (± 0.0079)	0.0236 (± 0.0029)	91
4	5000	1000	0.0187 (± 0.0016)	0.0195 (± 0.0011)	0.076 (± 0.002)	0.0171 (± 0.0016)	762
5	9000	1000	0.0178 (± 0.0012)	0.0172 (± 0.0013)	0.0706 (± 0.0037)	0.0158 (± 0.0017)	3618

Table 3.1: Experimental results for Laplace GPC. n : task training set size; n_{MS} : model selection training set size. emp: expected empirical error; gen: expected generalisation error (estimated as average over test set). upper: upper bound on expected generalisation error after Theorem 3.1. gen-bayes: test error of corr. Bayes classifier. time: total running time per run (secs). Figures are mean and width of 95% t -test confidence interval.

Note that the resource requirements for our experiments are well within today’s desktop machines computational capabilities. For example, experiment #4 was completed in total time of about 12 to 13 hours, the memory requirements are around 250M. Now, for this setting both the expected empirical error and the estimate (on the test set) of the expected generalisation error lie around 2%, while the PAC bound on the expected generalisation error given by Theorem 3.1 is 7.6% — an impressive, highly non-trivial result on samples of size $n = 5000$. Our largest experiment #5 was done mainly for comparison with experiment #2 for IVM (see Section 3.5.3). The total computation time was 6 hours for each iteration, and the memory requirements are around 690M. We note a slight improvement in test errors as well as in the upper bound values (which now lie

around 7%).

The “gen-bayes” column in Table 3.1 contains the test error that a *Bayes* classifier with the same approximate posterior as the Gibbs classifier attains (see Section 3.1.2). Note that it is not necessarily the best we could obtain for a Bayes classifier, because the model selection is done specifically for the Gibbs variant. In the Laplace GPC case we note that Bayes and Gibbs variants perform comparably well, although the Bayes classifier attains slightly better results and, as mentioned in Section 3.3.2, can be evaluated more efficiently. We include these results for comparison only: although the Gibbs result implies a bound on the generalisation error of the Bayes classifier (see Section 3.2.5), the link is too weak to render a sufficiently tight result.

3.5.3 Experiments with Sparse Greedy GPC

The sparse IVM approximation to GP binary classification is discussed in Section 4.4.1 and the corresponding PAC-Bayesian theorem in Section 3.3.3. In comparison with Laplace GPC, both training and evaluation are much faster now: $O(nd^2)$ and $O(d^2)$ respectively. The bound value can be computed in $O(nd^2)$. In our experiments here, the final active set size d was fixed *a priori*. Training was done in the same way as discussed in Section 4.8.1. For all experiments reported here, we chose MS training size $n_{\text{MS}} = 1000$, MS validation size $l_{\text{MS}} = 1000$ and $d_{\text{MS}} = 150$. Note that in experiments which have the same $(n, n_{\text{MS}}, l_{\text{MS}})$ constellation as Laplace GPC experiments, we use the same data subsets, in order to facilitate direct comparisons. The results are listed in Table 3.2.

#	n	d	emp	gen	upper	gen-bayes	time
1	5000	500	0.0154 (± 0.0021)	0.0207 (± 0.0015)	0.067 (± 0.0026)	0.0084 (± 0.0014)	16
2	9000	900	0.0101 ($\pm 6.88\text{e-}4$)	0.0116 ($\pm 5.49\text{e-}4$)	0.0502 ($\pm 6.13\text{e-}4$)	0.0042 ($\pm 8.79\text{e-}4$)	82

Table 3.2: Experimental results for sparse GPC. n : task training set size; d : final active set size. emp: expected empirical error; gen: expected generalisation error (estimated as average over test set). upper: upper bound on expected generalisation error after Theorem 3.1. gen-bayes: test error of corr. Bayes classifier. time: total running time per run (secs). Figures are mean and width of 95% t -test confidence interval.

Let us compare these results to the ones obtained for Laplace GPC. The sparse GPC Gibbs classifier trained with 5000 examples attains an expected test error of 2.1%, and the upper bound evaluates to 6.7%. While the former is the same as for the Laplace GPC variant, the latter is significantly lower. The ratio between

upper bound and expected test error is 3.19, the ratio between gap bound and expected test error is 2.46. Note that experiment #1 for the sparse GPC was completed in total time of about 16 minutes — almost fifty times faster than the Laplace GPC experiment #4. It is interesting to observe that for this sample size, the results here are significantly better than for the full Laplace GPC on the same task¹⁷ (experiment #5 in Section 3.5.2). Finally note that we did not try to optimise the final active set size d , but simply fixed $d = n/10$ *a priori*. An automatic choice of d could be based on heuristics which evaluate the error on the datapoints outside the active set (see Section 4.4.1).

The “gen-bayes” column in Table 3.2 serves the same purpose as the “gen-bayes” column in Table 3.1. In case of sparse greedy GPC, the results show that the Bayes classifier performs significantly better than the Gibbs variant, although the latter still attains very competitive results. A possible explanation for this difference, given that it cannot be observed for Laplace GPC, is obtained by inspecting the (C, w) kernel parameters values that are preferred by sparse greedy GPC. The parameter C is much larger for sparse GPC, i.e. the latent process has a larger *a priori* variance. This typically leads to an increase in the predictive variances, which in turn might introduce more sampling errors in the Gibbs predictions.

3.5.4 Comparison with PAC Compression Bound

In this section, we present experiments in order to compare our result for sparse GPC (Section 3.5.3) with a state-of-the-art PAC compression bound. Note that here, we employ *Bayes* GP classifiers instead of Gibbs GP classifiers: it would not be fair to compare our Gibbs-specific bound to an “artificially Gibbs-ified” version of a result which is typically used with Bayes-like “averaging” classifiers. A compression bound applies to learning algorithms which have some means of selecting a subsample S_I of size d from a sample of size n , such that the hypothesis they output is independent of $S_{\setminus I} = S \setminus S_I$, given S_I . More details are given in Appendix B.6, where we also state and prove the compression bound we are using here (Theorem B.2). Examples of compression schemes are the perceptron learning algorithm of [155], the support vector machine (SVM) and the sparse greedy IVM. The PAC compression bound of Theorem B.2 depends only on the training error for the remaining $n - d$ patterns $S_{\setminus I}$ outside the active set (called $\text{emp}^{\setminus d}(S)$) and on d . We repeated the experimental setup used in Section 3.5.3

¹⁷Note that we are comparing two quite different ways of approximating the true posterior by a Gaussian: a Laplace approximation around the *mode* (which is different from the posterior *mean* — the “holy grail” of Bayesian logistic regression, see Section 2.1.3) and an approximation based on repeated moment matching (see Section 4.3). A more meaningful direct comparison would involve the cavity TAP method of [139] (see Section 4.3) which is, however, more costly to compute than the Laplace approximation.

and employed the same dataset splits. The results can be found in Table 3.3.

#	n	d	emp	gen	upper
1	5000	500	0.0025 ($\pm 6.79\text{e-}4$)	0.0058 (± 0.0015)	0.3048 (± 0)
2	9000	900	0.0024 ($\pm 4.25\text{e-}4$)	0.003 ($\pm 7.54\text{e-}4$)	0.3041 (± 0)

Table 3.3: Experimental results for PAC compression bound with sparse GP Bayes classifier. n : task training set size; d : final active set size. emp: empirical error (on full training set); gen: error on test set. upper: upper bound on generalisation error given by PAC compression bound. Figures are mean and width of 95% t -test confidence interval.

For both experiments, $\text{emp}^d(S) = 0$ was achieved in all runs, the compression bound is tightest in this case. Nevertheless, in experiment #1, the upper bound on the generalisation error is 30.5%, a factor of 50 above our estimate on the test set. The ratio is even worse for experiment #2. At least on this task, the PAC-Bayesian theorem produces much tighter upper bound values than the PAC compression bound. Note also that the compression bound is more restrictive, in that it applies to compression schemes only. On the other hand, the PAC-Bayesian theorem depends much more strongly on the model, prior assumptions and inference approximation algorithm, while the compression bound cannot differentiate between algorithms which attain the same level of compression and empirical error $\text{emp}^d(S)$.

The reader may wonder why the generalisation errors here are slightly lower than the ones reported in Table 3.2. This should be due to the fact that in Section 3.5.3, we evaluated the Bayes classifier based on the hyperparameter values which have been optimised for the *Gibbs* variant, while here we performed model selection for the Bayes classifier explicitly.

3.5.4.1 Comparison with Compression Bound for Support Vector Classifiers

We can also compare our results for sparse GP Gibbs classifiers with state-of-the-art bounds for the popular soft-margin support vector machine (SVM). The latter is introduced in Section 2.1.6, where we also discuss similarities and differences to proper GP classification models. In the context here, it is important to note that the SVM algorithm often produces rather sparse predictors. However, the degree of sparseness is not a directly controllable parameter, furthermore it is not an explicit algorithmic goal of the SVM algorithm to end up with a maximally sparse expansion. The aim is rather to maximise the “soft” minimal empirical margin (see Section 2.1.6). SVM is a compression scheme (see Section B.6), where

d is the number of support vectors (vectors which are misclassified or lie within the margin tube around the decision hyperplane in feature space; see Section 2.1.6). Note that for SVM, we always have $\text{emp}^d(S) = 0$, since misclassified points are support vectors. Experimental results for SV classifiers and the PAC compression Theorem B.2 can be found in Table 3.4. Again, we employed the same dataset splits as used in Section 3.5.3.

#	n	emp	gen	upper	num-sv
1	5000	0.0016 ($\pm 9.49\text{e-}4$)	0.0048 (± 0.0012)	0.2511	370.8 (± 10.71)
2	9000	0.0021 ($\pm 7.67\text{e-}4$)	0.0036 (± 0.0012)	0.213	529 (± 29.12)

Table 3.4: Experimental results for PAC compression bound with SV classifiers. n : task training set size. emp: empirical error (on full training set); gen: error on test set. upper: upper bound on generalisation error given by PAC compression bound. Figures are mean and width of 95% t -test confidence interval.

In both experiments, a higher degree of sparsity is attained than the one chosen in the experiments above for sparse GPC (as mentioned above, we did not try to optimise this degree in the sparse GPC case), leading to somewhat better values for the PAC compression bound. However, the values of 25% (experiment #1) and 21% (experiment #2) are still by factors > 50 above the estimates computed on the test set, which is not useful in practice.

The compression bound applies to SVM, but is certainly not specifically tailored for this algorithm, since it does not even depend on the empirical margin distribution. Can we get better results with other SVM-specific bounds? The margin bound of [178], commonly used to justify data-dependent structural risk minimisation for SVM, becomes non-trivial (i.e. smaller than 1) only for $n > 34816$ (see [73], Remark 4.33). The algorithmic stability bound of [22] does not work well for support vector classification either. In fact, the gap bound value converges to zero at some rate $r(n)$ (for $n \rightarrow \infty$) only if the variance parameter¹⁸ C goes to zero at the same rate $r(n)$. If $r(n) = O(1/n)$ or $r(n) = O(\log n/n)$, this would correspond to severe over-smoothing. Herbrich and Graepel [73] use some older PAC-Bayesian theorems from [116] to prove a bound which depends on the minimal normalised empirical margin. This theorem applies to hard-margin SVC only and becomes non-trivial once the minimal normalised (hard) margin¹⁹ is > 0.91 , given that the feature space has dimension $> n$. Hard-margin SVMs

¹⁸In the SVM literature, it is common practice to separate C from the covariance kernel and write it in front of the sum over the slack variables. The parameter λ in [22] is $\lambda = 2/(Cn)$, and their gap bound behaves as $1/(n\lambda)$ as $n \rightarrow \infty$.

¹⁹If we view SVC as a linear method in a feature space induced by the covariance kernel, the

tend to overfit on noisy real-world data with very small normalised margins at least on some points, and in practice the soft-margin variant is typically preferred. In a separate experiment using the same setup and dataset splits as in #1 of this section (i.e. training sample size $n = 5000$), but training hard margin SVMs without bias parameter, we obtained generalisation error estimates on the test set of $\text{gen} = 0.0056 (\pm 3.904e-5)$, minimum normalised margins of $\text{minmarg} = 0.0242 (\pm 2.813e-5)$ and generalisation upper bound values of $\text{upper} = 16.28 (\pm 4.7e-3)$ using the theorem of [73]. These results back the simple observation that the minimum normalised (hard) margin is not suitable as a PAC gap bound statistic and should probably be replaced by one which is more robust against noise, such as soft margin, sparsity degree or combinations thereof. All in all, and much to our surprise, we were not able to find any proposed ‘‘SVC-specific’’ bound which would be tighter on this task than the simple PAC compression bound of Theorem B.2 used above.

3.5.5 Using the Bounds for Model Selection

Can our results be used for model selection? In our opinion, this issue has to be approached with care. It seems rather obvious that a generalisation error bound should not be used for model selection on a real-world task if it is very far above reasonable estimates of the generalisation error on this task. This argument is discussed in more detail in Section 2.2.4. Even though the PAC-Bayesian theorem applied to GP Gibbs classifiers offers highly non-trivial generalisation error guarantees for the real-world task described in this section, they still lie by a factor > 3 above the estimates on the test set. In spite of this fact, we follow the usual conventions and present results of an experiment trying to assess the model selection qualities of our bound.

Once more, we used sparse greedy GPC. The experiment consisted of six independent runs. We fixed $C = 120$ and used a grid of values for w . In each run, a sample S of size $n = 5000$ was drawn from the training pool, the method was trained for each configuration (C, w) (we fixed $d = 500$) and evaluated on the test set. The results are shown in Figure 3.3. We translated the upper bound values towards the expected test errors by subtracting a constant (determined as 95% of the average distance between points on the two curves). In each graph, the scale on the left hand side is for the expected test error, the scale on the right hand side for the upper bound value. In Figure 3.4, we plot expected test errors (horizontal) vs. upper bound values (vertical). In this type of plot a mostly

minimal normalised margin is the arc cosine of the maximal angle between the normal vector of the separating plane and any of the input points mapped into feature space. A minimal normalised margin close to 1 means that *all* mapped input points lie within a double cone of narrow angle around the line given by the normal vector. For noisy data, such a situation is arguably quite unlikely to happen.

monotonically increasing relationship is what we would ideally expect. The dotted curves are lines $x + b$ with slope 1, where b is fitted to the corresponding solid curves by minimum least squares. The ordering of the six subplots is the same as in Figure 3.3.

In this particular experiment, there is a surprisingly good monotonically increasing linear correlation between upper bound values and expected test errors, and model selection based on minimising the upper bound value might have worked in this case. However, note that the constants we had to subtract from the upper bound curves in order to bring them close to the expected generalisation error estimates for visual inspection, were an order of magnitude larger than the range of variation of the individual curves shown in the plots. An important future direction of research would be to gain more understanding, both empirically and analytically, of the phenomenon observed here: a slack term which is significantly larger than the expected test error, yet also much less variable than the latter over a range of hyperparameter values of interest (see also Section 3.2.6.2). Further experiments in Section 3.5.6 are however less conclusive and show that for other hyperparameters, the bound may not be predictive.

One might suspect that it is really only the expected empirical error which follows the expected test error closely, and that the difference between them remains fairly constant. After all, most of the points the empirical error is evaluated on are not in the active set. This argument of course ignores the fact that there is a dependence of the posterior on patterns outside the active set even for a com-

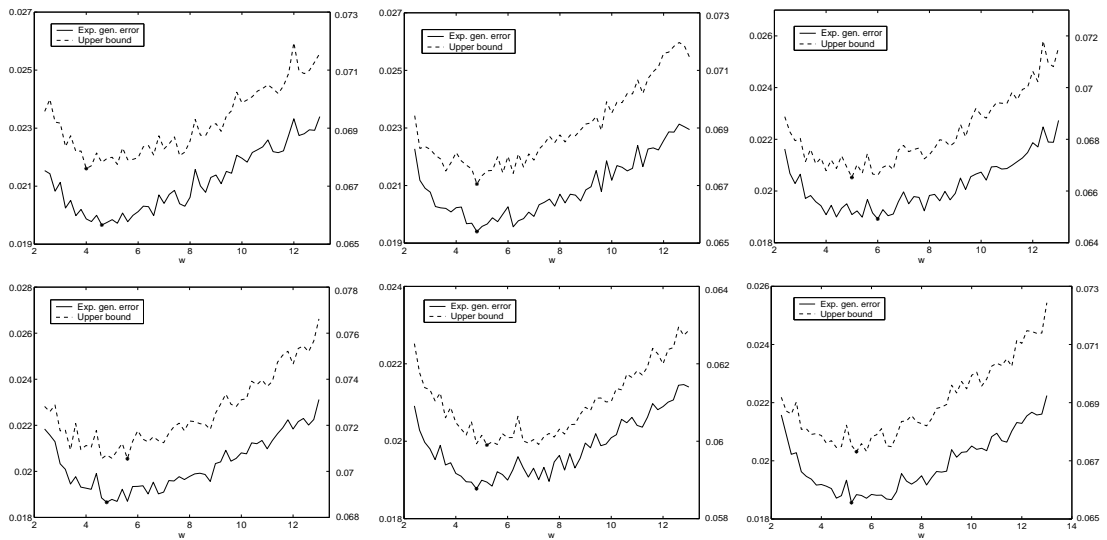


Figure 3.3: Comparing upper bound values with expected test errors. Solid line: expected test error (scale on left). Dashed line: upper bound value (*translated*, scale on the right). Respective minimum points marked by an asterisk.

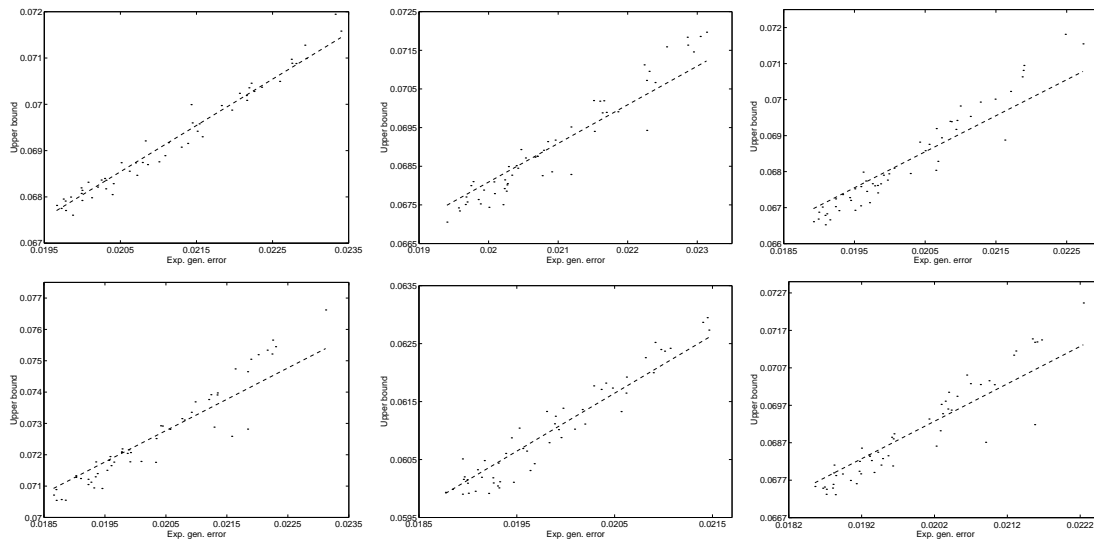


Figure 3.4: Comparing upper bound values (vertical axis) with expected test errors (horizontal axis). Dotted line: fitted regression line with slope 1.

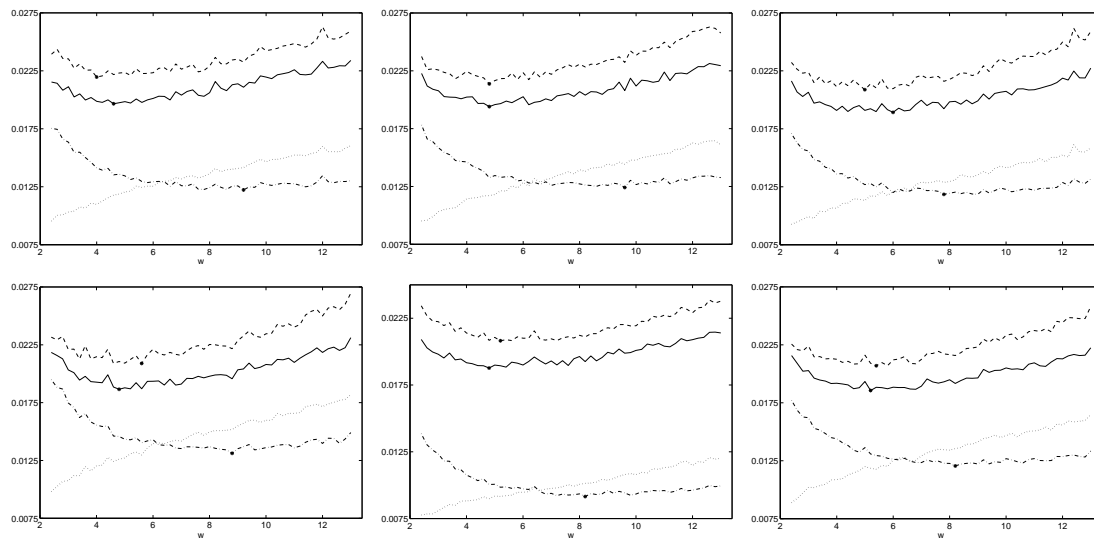


Figure 3.5: Comparing upper bound values with expected test errors (upper parts) and gap bound values with expected training errors (lower parts). Solid: expected test error (scale on left side). Dashed: upper bound value (*translated*). Dash-dotted: expected training error (scale on left side). Dotted: gap bound value (*translated*). Respective minimum points marked by an asterisk.

pression scheme such as the IVM, namely they have *not* been chosen in favour of the active set. In theory, a PAC bound tells us that such dependencies have to

be accounted for by an additional complexity term which is of rather crude union bound type in the PAC compression bound (Theorem B.2) and of a more refined type in the PAC-Bayesian theorem. By splitting the upper bound curves above into expected empirical errors and gap bound values, we see that this extension is indeed necessary even if $d \ll n$. In Figure 3.5 we plotted all these curves together in common graphs, using the same ordering of runs as in the other graphs. In each subplot, the two curves at the top are the upper bound (dashed) and the expected test error (solid), while the two curves at the bottom are the expected empirical error (dash-dotted) and the gap bound (dotted). The scale is correct for both the expected empirical and test error curves, while the gap and upper bound curves are translated downwards by different constants. The scale for the latter two curves is omitted. Furthermore, the graphs in Figure 3.6 show that the linear correlation between expected training and test errors is poor, and indeed most of the graphs in Figure 3.5 exhibit over-fitting: model selection based on minimising the expected training error would choose too large values of w , corresponding to a too narrow kernel width.

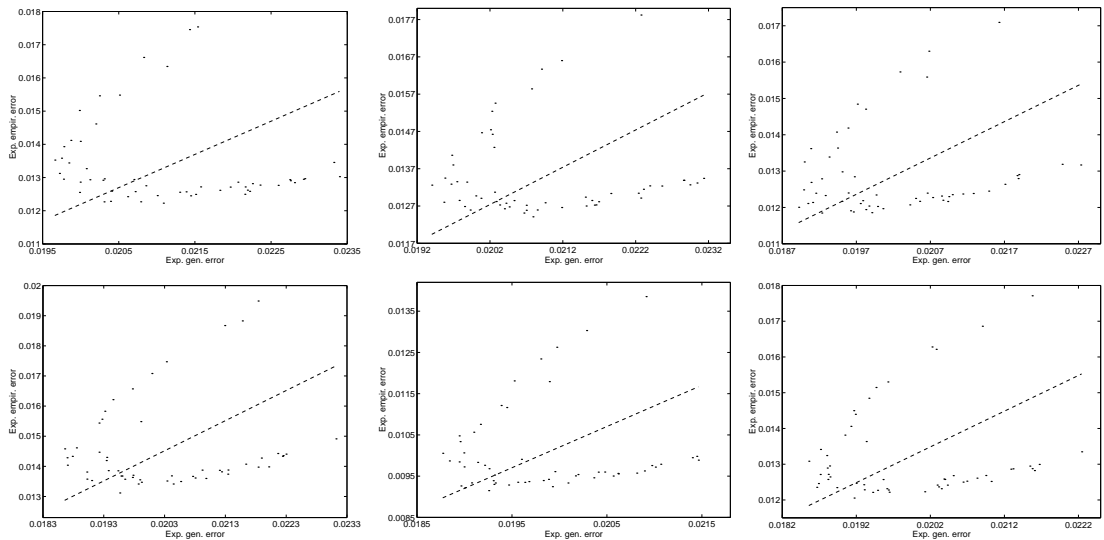


Figure 3.6: Comparing expected training errors (vertical axis) with expected test errors (horizontal axis). Dotted line: fitted regression line with slope 1.

3.5.6 Comparing Gibbs and Bayes Bounds

The aim of this section is twofold. First, in order to strengthen the results obtained in Sections 3.5.3 and 3.5.5 we repeat these experiments here using a different binary classification task. Second, we planned to compare the PAC-Bayesian theorem for Gibbs classifiers and its implications for the GP Bayes

classifier against Theorem 3.4 obtained by Meir and Zhang. However, it turned out that the latter theorem does not render non-trivial bounds on this task, highlighting the problem pointed out in Section 3.4.1.1.

The setup MNIST8/9 we used here is similar to MNIST2/3 introduced in Section 3.5.1, but differs in the following points. We now consider the MNIST subtask of discriminating eights from nines. We used all available images from the official training set in our training pool (11800 cases) and tested on all images from the official test set ($l = 1983$), and the full 28-by-28-pixel bitmaps served as input points here.

We repeated the experiments of Section 3.5.3 for the single training sample size $n = 8000$, fixing the active size to $d = 800$, furthermore $n_{\text{MS}} = l_{\text{MS}} = 1000$ and $d_{\text{MS}} = 150$. The outcome was $\text{emp} = 0.0105(\pm 6.93\text{e-}4)$, $\text{gen} = 0.0220(\pm 2.68\text{e-}4)$, $\text{upper} = 0.0568(\pm 0.0012)$, quite in line with the earlier figures.

In order to compare the two PAC-Bayesian theorems for the GP Bayes classifier, we re-ran these experiments, but now using the Bayes instead of the Gibbs variant for model selection.²⁰ As noted in Section 3.2.5, we can use twice the value of the Gibbs bound as an upper bound for the Bayes classifier, however this is unsatisfactory given the apparent superior performance of the Bayes classifier on this task. When we tried to compare these results against the bound of Theorem 3.4 of Meir and Zhang, we ran into the problem mentioned in Section 3.4.1.1: the term $4c\kappa^{-1}(2D[Q \parallel P]/n)^{1/2}$ was larger than one even for $\kappa = 1$, in all ten runs. In this situation, the bound cannot give non-trivial results for *any* prior configuration. We suspect that even in cases where the bound becomes non-trivial, the scaling with the *square root* of $D[Q \parallel P]/n$ will render it very sub-optimal in practice. Note that this task, with $n = 8000$, is at the upper range of sample sizes we considered in our experiments. From these experimental findings, we conclude that the problem of a practically satisfying PAC-Bayesian bound for GP Bayes classifiers is not resolved yet.

Given the results in Section 3.5.5, the PAC-Bayesian bound seems quite suitable for model selection due to an excellent monotonically increasing linear correlation between expected test errors and upper bound values for different values of w . We repeated these experiments on MNIST8/9 with $n = 8000$, resulting in Figure 3.7.

These results show that the upper bound value can have shortcomings as “predictor” for the expected test error. The plots in the left and middle columns are for fixed $C = 200$, $C = 100$ and varying w . Some values bail out of the otherwise linear relationship, notably the upper bound underestimates the sharp increase in test error for too small w . Recall that small w translate into large length scale and therefore very smooth discriminant functions (decision boundaries). A preference

²⁰The selection results were quite different. The Gibbs variant preferred larger C values, while the Bayes variant required larger w values.

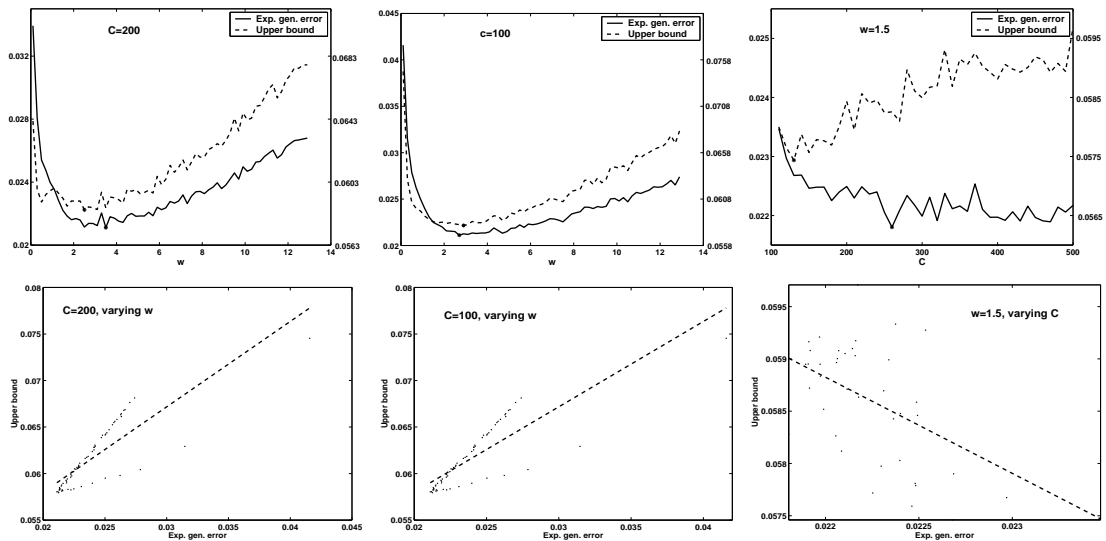


Figure 3.7: Comparing expected test errors with upper bound values on MNIST8/9, $n = 8000$. Upper row: hyperparameter values vs. test errors (scale left) and translated bound values (scale right). Lower row: test errors vs. bound values (dotted: fitted regression line). Left: varying w with $C = 200$. Middle: varying w with $C = 100$. Right: varying C with $w = 1.5$.

for over-smooth solutions hints towards the bound placing too much weight on the complexity penalty. The fact that for large w values the bound grows faster than the expected test error also points in this direction. The right column plots show that the upper bound fails to predict the test error for varying C and fixed $w = 1.5$, again favouring over-smooth solutions (C is the prior process variance). These findings somewhat back the objections we raise in Section 2.2.4, showing that model selection based on current PAC bounds is risky at best.

3.6 Discussion

In this chapter, we have shown how to apply McAllester’s PAC-Bayesian theorem to approximate Bayesian Gaussian process classification methods. The generic bound applies to a wide class of GP inference approximations, namely all that use a GP to approximate the predictive process (see Sections 2.1.3 and 4.6 for references). We have simplified the original proof of the PAC-Bayesian theorem considerably, pointing out convex duality (see Section A.3) as the principal underlying technique. We have also shown how to generalise the theorem to multi-class settings and the problem of bounding linear functions of the confusion matrix. Although in this work, we have focused on Gaussian process models, the PAC-Bayesian theorem can in principle be applied to any approximate Bayesian

classification technique. The terms the bound depends upon typically fall out as simple byproducts of the inference approximation.

Additional conclusions and suggestions for future work can be found in Section 5.1.