Gaussian Processes for Classification: Mean Field Algorithms

Manfred Opper
Neural Computing Research Group,
Department of Computer Science and Applied Mathematics,
Aston University, Birmingham B4 7ET, United Kingdom

Ole Winther
Theoretical Physics II, Lund University, Sölvegatan 14 A,
S-223 62 Lund, Sweden

CONNECT, The Niels Bohr Institute, Blegdamsvej 17,
2100 Copenhagen Ø, Denmark

September 27, 2000

To appear in Neural Computation vol. 12, issue 11 (2000)

Abstract

We derive a mean field algorithm for binary classification with Gaussian processes which is based on the TAP approach originally proposed in Statistical Physics of disordered systems. The theory also yields an approximate leave-one-out estimator for the generalization error which is computed with no extra computational cost. We show that from the TAP approach, it is possible to derive both a simpler 'naive' mean field theory and support vector machines (SVM) as limiting cases. For both mean field algorithms and support vectors machines, simulation results for three small benchmark data sets are presented. They show 1. that one may get state of the art performance by using the leave-one-out estimator for model selection and 2. the built-in leave-one-out estimators are extremely precise when compared to the exact leave-one-out estimate. The latter result is a taken as a strong support for the internal consistency of the mean field approach.

1 Introduction

Recently, there has been a great deal of interest in non-parametric Bayesian approaches to regression and classification which are based on the concept of Gaussian processes (Mackay, 1997; Williams, 1997; Williams & Rasmussen, 1997; Neal, 1997; Gibbs & Mackay, 1997; Barber & Williams, 1997; Williams & Barber, 1997; Opper & Winther, 1999a&b). The underlying idea is conceptually very simple. Instead of defining prior distributions over parameters of a learning machine (e.g. weights and biases in case of a neural net), one directly defines a Gaussian prior distribution over the *function* which the machine computes.

For regression problems, this approach allows for an explicit statistical inference by analytical methods. On the other hand, for non-linear statistical models like the ones used for classification, the high dimensional integrals which occur in performing posterior averages can only be treated by approximative methods. An approximation to these integrations can be based on Monte Carlo sampling (Neal, 1997) which, for a large number of data may be time consuming. Other, semi-analytic approaches use Laplace's methods, i.e. the approximation of the posterior by a multivariate Gaussian (Barber & Williams, 1997; Williams & Barber, 1997) or a tractable variational bound on the data likelihood (Gibbs & Mackay, 1997).

This paper deals with a different approach which has its origin in the Statistical Physics of disordered media. Their thermodynamic properties can be calculated from high dimensional expectations over Gibbs distributions which contain a large set of random quantities, similar to the input data in the posterior distribution of Bayesian analysis. Our method is based on a TAP (Thouless, Anderson & Palmer, 1977; Mézard, Parisi & Virasoro, 1987) type of mean field approximation, which goes beyond a variational approximation of the posterior by a product distribution. The TAP approach is a controlled approximation in the sense that it becomes exact for simple distributions of the disorder (the input data) in the limit where the number of integration variables approaches infinity (Mézard, Parisi & Virasoro, 1987; Opper & Winther, 1996). A second advantage is that the method, by its construction, automatically computes a leave-one-out estimate of its generalization error.

Clearly, for real data the distribution of the disorder is unknown and one cannot assess the validity of the mean field approach directly. We will therefore–in contrast to most prior applications of the TAP method–not make an explicit assumptions for this distribution. Instead, we base our approach on a Gaussian assumption for the so called *cavity fields* (Mézard, Parisi & Virasoro, 1987)–which we believe to be reasonably good for a large class of distributions–from which a closed set of non-linear equations for posterior averages can be derived. An alternative derivation of these equations was

given in our previous NIPS paper (Opper & Winther, 1999a) which used a less intuitive approximation for the Gibbs free energy (Parisi & Potters, 1995). We also show that the internal consistency of the mean field approximation can be checked indirectly through comparing the leave-one-out estimate for the generalization error provided by the TAP theory with the exact leave-one-out estimate.

The paper is organized as follows: Section 2 gives the basic definitions for classification with Gaussian processes. In Section 3, we derive the TAP mean field algorithm. A recipe for solving the mean field equations by iteration is given in Section 4. A leave-one-out (loo) estimator for generalization error of the TAP mean field algorithm is proposed in Section 5. In Section 6, we derive a 'naive' mean field algorithm. In 7, Support vector machines (SVMs) are obtained from TAP mean field theory by a limiting procedure in the prior and a change in the Likelihood (which ruins the probabilistic interpretation). In section 8, we present simulation results for three small benchmark data sets 'Sonar – Mines versus Rocks', 'Leptograpsus Crabs' and 'Pima Indians Diabetes'. The paper is concluded in Section 9. Appendices A and B discuss respectively the noise model and the covariance functions used.

2 Gaussian Process Models

We consider the following supervised learning problem: A training set, $D_N = \{(\mathbf{x}_i, t_i)|i = 1, \ldots, N\}$ of input vectors $\mathbf{x}_i$ and associated outputs t_i is given and we want to infer the output t for a new input $\mathbf{x}$, where we restrict ourselves to binary classification $t = \pm 1$.

The probability of the output t, at a given point $\mathbf{x}$, $p(t|h(\mathbf{x}))$ is assumed to depend on the input $\mathbf{x}$ through a scalar activation function $h(\mathbf{x})$, which must be inferred by the learning process. The simplest example for such a likelihood is the noise-free case

$$p(t|h(\mathbf{x})) = \Theta(t \ \mathrm{sgn} h(\mathbf{x})) = \Theta(t \ h(\mathbf{x})) \ , \tag{1}$$

where $\Theta(x) = 1$ for $x > 0$ and 0 otherwise. Thus, only functions that have the same sign as the training label have non-zero probability. A more common choice for the likelihood which corresponds to noisy or ambiguous classifications would replace eq. (1) with a *sigmoidal* function of $t\, h(\mathbf{x})$. Since our approach requires integrations of likelihood functions over Gaussian distributions, we will restrict ourselves to likelihoods corresponding to a *probit model*

$$p(t|h) = \Phi\left(\frac{th}{\sqrt{v_2}}\right) . \tag{2}$$

where Φ is the error function (or more precisely the Gaussian cumulative distribution function)

$$\Phi(x) = \int_{-\infty}^{x} \frac{dy}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \tag{3}$$

for which such integrations can be performed exactly. However, as discussed in Appendix A, the Bayesian classification for such a choice can also be derived from a model of the form eq. (1) by a simple redefinition which amounts to modify the prior over $h(\mathbf{x})$ such that it includes the noise. In the subsequent discussion we will therefore only consider the Likelihood eq. (1).

There are many possible parametric approaches to modeling the function $h(\mathbf{x})$. A popular one, e.g., is to represent h by a neural network and try to estimate the parameters from the data. In the Bayesian approach to learning, a prior probability distribution over the parameters (e.g. the weights of the network) which determine h is specified. To go one step further, one may skip the parameters and introduce a prior distribution over the entire space of functions h. The choice of a Gaussian probability measure over functions has been motivated from a study of the limiting prior distribution in the neural neural network case, when the number of hidden units grows large (Neal, 1996; Williams, 1997). In such a case, a typical function h drawn at random from the prior distribution is a realization of a Gaussian process.

A Gaussian process is a family of random variables $h(\mathbf{x})$, $\mathbf{x} \in T$, where T denotes a possibly uncountable index set, such that any finite collection $\mathbf{h} = (h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N))$ of random variables have a joint Gaussian distribution

$$p(\mathbf{h}) = \frac{1}{\sqrt{(2\pi)^N \det \mathbf{C}}} \exp\left(-\frac{1}{2}(\mathbf{h}-\mathbf{m})^T \mathbf{C}^{-1} (\mathbf{h}-\mathbf{m})\right) \ , \tag{4}$$

where $\mathbf{m}$ is the mean $\mathbf{m} = E(\mathbf{h})$ (which we will set to zero in the following) and $\mathbf{C} \equiv E(\mathbf{h}\mathbf{h}^T) - \mathbf{m}\mathbf{m}^T$ is the covariance matrix of the input set with elements $C(\mathbf{x}_i, \mathbf{x}_j)$. The function $C(\mathbf{x}, \mathbf{x}')$ explicitly determines how the activations at different points are correlated, which should reflect our prior beliefs about the variability of the function $h(\mathbf{x})$. In principle, we may choose any positive semi-definite function for C. Several parametric choices, some of them related to neural network models, are discussed in Appendix B.

Formally, we may represent a Gaussian process model by a single layer neural network with (possibly infinitely many) weights w_ρ, $\rho = 1, 2, \ldots$ with a spherical Gaussian prior distribution $E[w_\rho w_{\rho'}] = \delta_{\rho\rho'}$, by performing a *Karhunen-Loeve* expansion, see e.g. (Papoulis, 1994)

$$h(\mathbf{x}) = \sum_\rho w_\rho \sqrt{\lambda_\rho} \Psi_\rho(\mathbf{x}) \tag{5}$$

where $\Psi_\rho(\mathbf{x})$ are eigenfunctions of the covariance function (kernel) C with

eigenvalues λ_ρ. From eq. (5), we get the kernel representation $C(\mathbf{x}, \mathbf{x}') = \sum_\rho \lambda_\rho \Psi_\rho(\mathbf{x}) \Psi_\rho(\mathbf{x}')$.

At first glance, the infinite dimensional nature of the Gaussian process $h(\mathbf{x})$ seems a bit awkward. However, since we we are interested in making inference on a single input $\mathbf{x}$ based on a discrete set of training data inputs $\mathbf{x}_i$, one only needs the $N+1$ dimensional Gaussian prior distribution $p(\mathbf{h}, h)$ where $\mathbf{h} = h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N)$ and $h = h(\mathbf{x})$ is the random variable associated with the new input $\mathbf{x}$.

Using Bayes rule together with the likelihood eq. (1), one may now form the joint posterior distribution of $\mathbf{h}$ and h

$$p(\mathbf{h}, h|\mathbf{t}) = \frac{1}{p(\mathbf{t})} \prod_{i=1}^{N} p(t_i|h(\mathbf{x}_i))\; p(\mathbf{h}, h) = \frac{1}{p(\mathbf{t})} p(\mathbf{t}|\mathbf{h})\; p(\mathbf{h}, h)\;, \tag{6}$$

where $\mathbf{t} = t_1, \ldots, t_N$ is shorthand for the training set outputs and

$$p(\mathbf{t}) = \int d\mathbf{h} dh\; p(\mathbf{t}|\mathbf{h}) p(\mathbf{h}, h) = \int d\mathbf{h}\; p(\mathbf{t}|\mathbf{h}) p(\mathbf{h}) \tag{7}$$

is a normalization constant. To find $p(h|\mathbf{t})$ formally all we need is to marginalize over $\mathbf{h}$

$$p(h|\mathbf{t}) = \int d\mathbf{h}\; p(\mathbf{h}, h|\mathbf{t})\;. \tag{8}$$

We will call $p(h(\mathbf{x})|\mathbf{t})$ the *predictive* posterior of h. The term predictive is used because $\mathbf{x}$ is not in the training set and this posterior may–as will be shown below–be used for making Bayesian predictions. The predictive posterior is also a central concept in our mean field theory since the mean field approximation amounts to assuming a special simple parametric form of this distribution.

2.1 Bayesian Classification

The predictive posterior $p(h(\mathbf{x})|\mathbf{t})$ summarizes the knowledge about $h(\mathbf{x})$ after having observed the training set. We may calculate the *predictive probability* of the output label

$$p(t|\mathbf{t}) = \langle p(t|h) \rangle \equiv \int dh p(t|h) p(h|\mathbf{t})\;, \tag{9}$$

where we have introduced the notation $\langle \ldots \rangle$ for a posterior average. Bayes algorithm for classification selects the output with highest probability

$$t^{\text{Bayes}}(D_N, \mathbf{x}) = \operatorname*{argmax}_t p(t|\mathbf{t})$$

in order to minimize the posterior probability of an error. For binary classification, Bayes algorithm becomes

$$\begin{aligned} t^{\text{Bayes}}(D_N, \mathbf{x}) &= \text{sgn}\left[p(+1|\mathbf{t}) - p(-1|\mathbf{t})\right] \\ &= \text{sgn}\langle \text{sgn} h\rangle \ . \end{aligned} \tag{10}$$

In the Bayesian framework, we may also quantify the uncertainty of the prediction of Bayes algorithm. According to our posterior belief, the probability for the the output t to be correct is given by the predictive probability $p(t|\mathbf{t})$. The estimate of the probability that Bayes algorithm gives the wrong answer is therefore $p(-t^{\text{Bayes}}(D_N, \mathbf{x})|\mathbf{t})$.

3 Mean Field Theory

The posterior average needed to derive Bayes algorithm is in most cases of interest not analytically tractable. For the likelihood eq. (1), one has to resort to approximate techniques. In this paper we introduce an advanced mean field theory approach based on ideas of statistical mechanics.

The basic idea of mean field theories is to approximate the statistics of a random variable which is correlated to other random variables by assuming that the influence of the other variables can be compressed into a single effective mean 'field' with a rather simple distribution. Often, such an approximation is based on a variational product approximation for the distribution of interest, which neglects the correlation between random variables (Parisi, 1988). For applications to Gaussian processes, see (Csató *et. al.*, 1999) and in graphical models, see (Jordan, 1999). Our approach goes beyond such a simple mean field theory and is equivalent to the so called TAP mean field theory known in Statistical Mechanics of disordered systems.

We assume that given the data, the posterior distribution of h at a new data point $\mathbf{x}$, which was not in the set of training inputs, the predictive posterior, can be approximated by a Gaussian with mean $\langle h\rangle$ and variance $\lambda = \langle h^2\rangle - \langle h\rangle^2$: that is

$$p(h(\mathbf{x})|\mathbf{t}) \approx \frac{1}{\sqrt{2\pi\lambda}} \exp\left(-\frac{(h - \langle h\rangle)^2}{2\lambda}\right). \tag{11}$$

To motivate such an approximation, one may look at the expansion eq. (5). Clearly, the posterior distribution of the weight vector, $p(w_1, w_2, \ldots|\mathbf{t})$ will not be Gaussian for a non–Gaussian likelihood. However, one might justify a Gaussian approximation by the central limit theorem if (a) the weights w_ρ are sufficiently weakly dependent, and if (b) in addition the fluctuations of the sum (5) were dominated by a large number of terms with roughly the same

magnitude. One may argue that condition (b) will not be fulfilled when the eigenvalues λ_ρ are rapidly decreasing with ρ (e.g. exponentially fast). However, when the dimension of the input space is large and we assume a kernel which is permutation invariant with respect to the components of the input vector, there will be many features (e.g. the linear ones) which have the *same* eigenvalues and may thus give similar contributions to the fluctuations of eq. (5). The quality of the Gaussian approximation to $h(\mathbf{x})$ will strongly depend on the input $\mathbf{x}$. If we take a point which is close to one of the training inputs $\mathbf{x} \approx \mathbf{x}_i$, and e.g. consider the likelihood eq. (1) which imposes the strict inequality $t_i h(\mathbf{x}_i) > 0$, the distribution of $h(\mathbf{x})$ should be rather different from a Gaussian (see the subsequent discussion after eq.(15)). However, as we will see in a moment, the Gaussian approximation will be only applied to test points $\mathbf{x}$ and to training inputs $\mathbf{x}_i$, for which the label t_i and the corresponding likelihood factor has been discarded. In such a case, we expect that the approximation should be fairly good, when input points are typically not close to each others. In fact, for specific models (Opper & Winther 1996), the approximation becomes exact when the dimension of the input space approaches infinity and the input vectors are drawn at random from a probability distribution with independent components. In this limit, with probability one, inputs vectors will come out uncorrelated.

In the theory of disordered media, similar approximations are made for the distribution of the magnetic field at the cavity which is left, when an atom (corresponding to the data label) is removed from the system. Hence, we will also speak of cavity fields or cavity distributions. For more detailed discussion of the validity of the approximation for specific models in the Statistical Physics of disordered media, see (Mézard, Parisi & Virasoro, 1987).

We will show in the following, that means and variances for the Gaussian approximation eq. (11) can be calculated in a self-consistent way. Before doing so, we use the Gaussian approximation to calculate the predictive probability eq. (9)

$$p(t|\mathbf{t}) = \Phi\left(t\frac{\langle h(\mathbf{x})\rangle}{\sqrt{\lambda}}\right) \ , \tag{12}$$

where Φ is the error function eq. (3). The Bayes classifier becomes

$$t^{\text{Bayes}}(\mathbf{x}, D_N) = \text{sgn}\langle h(\mathbf{x})\rangle \ . \tag{13}$$

Note, that this result holds for any distribution which is symmetric around the mean value. The approximation to the error probability may also be calculated

$$p(-t^{\text{Bayes}}(\mathbf{x}, D_N)|\mathbf{t}) = \Phi\left(-\frac{|\langle h\rangle|}{\sqrt{\lambda}}\right) \ . \tag{14}$$

It is important to note that the Gaussian mean field approximation to the predictive posterior is not equivalent to a Gaussian approximation to the full

posterior. The latter approximation is commonly used in Bayesian modeling for parametric models, in the limit where the number of data is much larger than the number of parameters. This would not be justified in our non-parametric case. Second, it would also not be justified for the non-smooth likelihoods eq. (1). To understand the difference to this simpler approximation, consider the case where a $N+1$th example $(\mathbf{x}, t)$ has been observed. Then–according to Bayes rule–the posterior distribution is given by

$$p(h(\mathbf{x})|\mathbf{t}, t) = \frac{p(t|h(\mathbf{x}))p(h(\mathbf{x})|\mathbf{t})}{\int dh\; p(t|h)p(h|\mathbf{t})} \; . \tag{15}$$

For the likelihood eq. (1), the posterior distribution is far from being Gaussian as illustrated in figure 1. To derive the mean field algorithm, we shall exploit that–under the Gaussian approximation–one can derive an analytical relation between the mean of predictive distribution, $\langle h \rangle$ and the mean of $p(h(\mathbf{x})|\mathbf{t}, t)$.

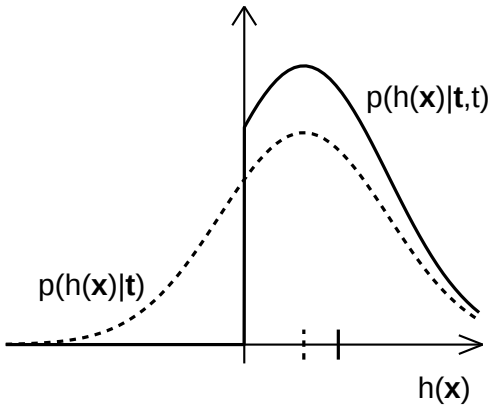

Figure 1: The predictive posterior of $h(\mathbf{x})$, $p(h(\mathbf{x})|\mathbf{t})$ (dashed line) and the posterior $p(h(\mathbf{x})|\mathbf{t}, t)$ (solid line) after the addition of the $N+1$th example $(\mathbf{x}, t)$. In the figure $t = 1$. The lines crossing the x-axis indicates the mean value of the two distributions.

Within the mean field approximation we thus need to derive expressions for $\langle h \rangle$ and $\lambda = \langle h^2 \rangle - \langle h \rangle^2$ to make Bayesian predictions and quantify the uncertainty of the prediction. This will be the task of the remainder of this section. We will start out by deriving exact expressions for the two first moments of the fields. The exact expressions are written in terms of averages over cavity distributions (for which one of the training examples is left out of the likelihood). These averages may be carried out within the mean field approximation and the second moments of the cavity distribution are determined self-consistently. The final result is a set of $2N$ non-linear mean field equations which has to be solved to find $\langle h \rangle$ and λ. In simulations this will be done by iteration.

The Gaussian assumption for the distribution of the cavity field is only one possibility to derive the TAP mean field theory. Another derivation which is based on a perturbation expansions for the Gibbs free energy (Opper & Winther, 1999a) is less intuitive and requires a somewhat deeper background of specific techniques of Statistical Physics. Such alternative derivations will be discussed in more detail elsewhere.

Exact expressions for $\langle h \rangle$ and $\langle h(\mathbf{x}_i) \rangle$. The starting point of our approximative treatment is based upon exact relations for the posterior mean

$$\langle h \rangle = \frac{1}{p(\mathbf{t})} \int d\mathbf{h} dh \, h \, p(\mathbf{t}|\mathbf{h}) p(\mathbf{h}, h)$$

and the posterior variance. The exact relations are derived using

$$\mathbf{h} p(\mathbf{h}) = \mathbf{C}\mathbf{C}^{-1}\mathbf{h} p(\mathbf{h}) = -\mathbf{C}\frac{\partial}{\partial \mathbf{h}} p(\mathbf{h}) \tag{16}$$

which follows from eq. (4) (with $\mathbf{m} = 0$). We may now find the exact relation for the posterior mean $\langle h(\mathbf{x}) \rangle$ at an arbitrary point $\mathbf{x}$ by extending the prior to include the new point $\mathbf{x}$ ($h(\mathbf{x})$ is the $N+1$th field) and applying integration by parts to shift the differentiation from the prior to the Likelihood:

$$\begin{aligned} \langle h(\mathbf{x}) \rangle &= -\frac{1}{p(\mathbf{t})} \int d\mathbf{h} dh \, p(\mathbf{t}|\mathbf{h}) \sum_{i=1}^{N+1} C(\mathbf{x}, \mathbf{x}_i) \frac{\partial}{\partial h_i} p(\mathbf{h}, h) \\ &= \sum_{i=1}^{N} C(\mathbf{x}, \mathbf{x}_i) \frac{1}{p(\mathbf{t})} \int d\mathbf{h} dh \, p(\mathbf{h}, h) \frac{\partial}{\partial h_i} p(\mathbf{t}|\mathbf{h}) \\ &= \sum_{i=1}^{N} C(\mathbf{x}, \mathbf{x}_i) \, t_i \, \alpha_i \ . \end{aligned} \tag{17}$$

Here we have introduced the notation $h_i \equiv h(\mathbf{x}_i)$ and defined the 'embedding strength'

$$\alpha_i \equiv \frac{t_i}{p(\mathbf{t})} \int d\mathbf{h} p(\mathbf{h}) \frac{\partial}{\partial h_i} p(\mathbf{t}|\mathbf{h}) \ . \tag{18}$$

Introducing the further shorthand notation $C_{ij} = C(\mathbf{x}_i, \mathbf{x}_j)$ and applying eq. (17) to the training data points $\mathbf{x} = \mathbf{x}_i$, we have

$$\langle h_i \rangle = \sum_j C_{ij} t_j \alpha_j \ . \tag{19}$$

These exact expressions are very interesting, because they allow us to express the expected activation $\langle h(\mathbf{x}) \rangle$ as a weighted average over the examples of the training set, as it is also the case in the support vector learning approach

(Vapnik, 1995) and in the MAP approach to Gaussian processes (Williams & Barber, 1998). In calculating the activation at a point $\mathbf{x}$, the training data are weighted according to how close they are to $\mathbf{x}$ (in an appropriate metric) by the kernel function, which defines the relevant neighborhood for each point in the input space. The overall importance of a training point $\mathbf{x}_i$ in the learning problem is measured by the variable α_i which as it may be seen from eq. (18) is always non-negative when $p(t|h)$ is an increasing function of ht.

For our cavity derivation, it is useful to introduce a new set of *predictive posteriors*, one for each example, by

$$p(h_i|\mathbf{t}\backslash t_i) = \frac{\int \prod_{j\neq i} dh_j \prod_{j\neq i} p(t_j|h_j)p(\mathbf{h})}{\int d\mathbf{h} \prod_{j\neq i} p(t_j|h_j)p(\mathbf{h})} \ . \tag{20}$$

where $\mathbf{t}\backslash t_i = t_i, \ldots, t_{i-1}, t_{i+1}, \ldots, t_N$ denotes a training set without the ith example. Denoting an average over this distribution by

$$\langle \ldots \rangle_{\backslash i} = \int dh_i \ldots p(h_i|\mathbf{t}\backslash t_i) \ ,$$

we can rewrite eq. (18) as

$$\alpha_i = t_i \frac{\langle \frac{\partial}{\partial h_i} p(t_i|h_i) \rangle_{\backslash i}}{\langle p(t_i|h_i) \rangle_{\backslash i}} \ . \tag{21}$$

Mean field expression for α_i The main reason why it is possible to calculate averages over $p(h_i|\mathbf{t}\backslash t_i)$ is the fact that it is a predictive posterior of the field at an input $\mathbf{x}_i$, i.e. t_i is not included in the data set. We can therefore apply the same Gaussian approximation as in eq. (11)

$$p(h_i|\mathbf{t}\backslash t_i) \approx \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left(-\frac{(h_i - \langle h_i \rangle_{\backslash i})^2}{2\lambda_i} \right) \tag{22}$$

with the variance defined as $\lambda_i = \langle h_i^2 \rangle_{\backslash i} - \langle h_i \rangle_{\backslash i}^2$. Inserting (22) into (21) we derive the explicit expression

$$\alpha_i \approx \frac{1}{\sqrt{\lambda_i}} \frac{D\left(\frac{\langle h_i \rangle_{\backslash i}}{\sqrt{\lambda_i}}\right)}{\Phi\left(t_i \frac{\langle h_i \rangle_{\backslash i}}{\sqrt{\lambda_i}}\right)} \ , \tag{23}$$

where we have introduced the Gaussian measure

$$D(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \ .$$

So far we have accomplished to write α_i in terms of new unknown quantities, the two first moments of the predictive distribution for h_i: $\langle h_i \rangle_{\backslash i}$ and λ_i.

We therefore need to express these in terms of some known quantities. For the first moments, we may exploit the following exact relation between the average over the posterior and the predictive posterior

$$\langle \ldots \rangle = \frac{\langle \ldots p(t_i|h_i) \rangle_{\setminus i}}{\langle p(t_i|h_i) \rangle_{\setminus i}} \ . \tag{24}$$

Since $p(h_i|\mathbf{t}\setminus t_i)$ is (assumed to be) Gaussian, we get (following the same reasoning as eq. (16): $h_i\, p(h_i|\mathbf{t}\setminus t_i) \approx \langle h_i \rangle_{\setminus i} - \lambda_i \frac{\partial}{\partial h_i} p(h_i|\mathbf{t}\setminus t_i)$ and

$$\begin{aligned} \langle h_i \rangle &= \langle h_i \rangle_{\setminus i} - \lambda_i \frac{\int dh_i p(t_i|h_i) \frac{\partial}{\partial h_i} p(h_i|\mathbf{t}\setminus t_i)}{\langle p(t_i|h_i) \rangle_{\setminus i}} \\ &= \langle h_i \rangle_{\setminus i} + \lambda_i \frac{\langle \frac{\partial}{\partial h_i} p(t_i|h_i) \rangle_{\setminus i}}{\langle p(t_i|h_i) \rangle_{\setminus i}} \ . \end{aligned} \tag{25}$$

Comparing with eq. (21) we see that

$$\langle h_i \rangle \approx \langle h_i \rangle_{\setminus i} + \lambda_i t_i \alpha_i \ . \tag{26}$$

This relation shows that adding the ith example to the posterior gives a change in the mean of the field in the direction of the target as may also be seen from fig. 1. Note from eqs. (23) and (26) that the equations for α_i are non-linear and thus have to be solved numerically. In sec. 5, we will exploit the fact that we have calculated $\langle h_i \rangle_{\setminus i}$ for every training example $i = 1, \ldots, N$ to propose a mean field estimate of leave-one-out estimator for the generalization error.

Mean field expressions for λ and λ_i. To complete our mean field theory, we have to derive expressions for the second moments: $\lambda = \langle h^2(\mathbf{x}) \rangle - \langle h(\mathbf{x}) \rangle^2$ and $\lambda_i \equiv \langle h_i^2 \rangle_{\setminus i} - \langle h_i \rangle_{\setminus i}^2$. The derivation is given in appendix C. The final results are:

$$\lambda \approx C(\mathbf{x}, \mathbf{x}) - \sum_{jk} C(\mathbf{x}, \mathbf{x}_j) \left[(\mathbf{\Lambda} + \mathbf{C})^{-1} \right]_{jk} C(\mathbf{x}_k, \mathbf{x}) \tag{27}$$

$$\lambda_i \approx \frac{1}{[(\mathbf{\Lambda} + \mathbf{C})^{-1}]_{ii}} - \Lambda_i \tag{28}$$

with $\mathbf{\Lambda} = \mathrm{diag}(\Lambda_1, \ldots, \Lambda_N)$ and

$$\Lambda_i \equiv -\lambda_i - t_i \left(\frac{\partial \alpha_i}{\partial \langle h_i \rangle_{\setminus i}} \right)^{-1} \ . \tag{29}$$

An explicit expression for $\frac{\partial \alpha_i}{\partial \langle h_i \rangle_{\setminus i}}$ is with some algebra obtained from eq. (23)

$$\frac{\partial \alpha_i}{\partial \langle h_i \rangle_{\setminus i}} \approx -\frac{\alpha_i}{\lambda_i} \langle h_i \rangle \ . \tag{30}$$

Eq. (28) with eqs. (29) and (30) thus defines a set of non-linear equations for λ_i, $i = 1, \ldots, N$.

For Gaussian process regression with a Gaussian noise model, one finds the same expressions for λ and λ_i with Λ_i equal to the noise variance (Williams & Rasmussen, 1996). It is therefore tempting to interpret Λ_i as the 'effective' noise variance for the ith example.

Summary of mean field equations We conclude the derivation of the TAP mean field equations with a summary of the results: The mean field prediction for new data point $\mathbf{x}$ follows from eqs. (13) and (17)

$$t^{\text{Bayes}}(\mathbf{x}, D_N) \approx \text{sgn} \sum_i C(\mathbf{x}, \mathbf{x}_i) t_i \alpha_i \ . \tag{31}$$

To obtain α_i, $i = 1, \ldots, N$, one has to solve the set of $2N$ non-linear mean field equations for α_i and λ_i, where α_i is given by eq. (23) (with eqs. (19) and (26)) and λ_i is given by eq. (28) (with eqs. (29) and (30)). In the next section we give a recipe for solving the mean field equations.

4 Solving the Mean Field Equations

The mean field equations are solved by iteration. The parallel iterative scheme used are described below in pseudo-code, where we also indicated which equations are used. The iterative scheme for the naive mean field (see section 6) is obtained by omitting the statements for Λ_i and λ_i in the iteration step.

Computationally, the most expensive step of the TAP mean field approach is the inversion of the $\mathbf{C}+\mathbf{\Lambda}$-matrix which is of $\mathcal{O}(N^3)$ compared to $\mathcal{O}(N^2)$ for the other operations. It turns that it requires many more iterations to solve for the equations for α_i than for λ_i which seems to be somewhat insensitive to precise value of the α_is. We therefore choose to make a (greedy) update of λ_i only every 20th iteration step.

For the problems studied convergence to the required accuracy could be obtained in less than 100 iteration steps and takes about 1 CPU second on an Alpha 433au for the largest problem studied ($N = 200$).[1]

Initialization.

1. Learning rate $\eta := 0.05$ and fault tolerance ftol $:= 10^{-4}$.

[1] An important contributing factor to learning speed is the use of an iterative learning rate: We set $\eta := 1.1\eta$ if 'the error' $\sum_i |\delta\alpha_i|^2$ decreases in the update step and $\eta := \eta/2$ otherwise.

2. Calculate the covariance matrix $\mathbf{C}$ (see appendix B).

3. For every training example i: $\alpha_i := 0$ and $\lambda_i := C_{ii}$.

Iterate. While $\max_i |\delta\alpha_i|^2 >$ ftol do:

1. For every i: (Eqs. (23,19,26))

$$\begin{aligned} \langle h_i \rangle &:= \sum_j C_{ij} t_j \alpha_j \\ \delta\alpha_i &:= \frac{1}{\sqrt{\lambda_i}} \frac{D(z_i)}{\Phi(z_i)} - \alpha_i, \qquad z_i \equiv t_i \frac{\langle h_i \rangle - \lambda_i t_i \alpha_i}{\sqrt{\lambda_i}} \; . \end{aligned}$$

2. For every i: $\alpha_i := \alpha_i + \eta\delta\alpha_i$

3. For every 20th iteration:

   (a) For every i: $\Lambda_i := \lambda_i \left(\frac{1}{\langle h_i \rangle t_i \alpha_i} - 1 \right)$. (Eqs. (29,30)).

   (b) For every i: $\lambda_i := \frac{1}{[(\mathbf{\Lambda}+\mathbf{C})^{-1}]_{ii}} - \Lambda_i$. (Eq. (28)).

5 Leave-One-Out Estimator

Once we have solved the mean field equations, the cavity average of the activation function $\langle h_i \rangle_{\setminus i}$ will be known. We may use these to define a leave-one-out (loo) estimator for the generalization error of the algorithm since $\mathrm{sgn}\langle h_i \rangle_{\setminus i}$ is a mean field estimate of the prediction of the algorithm on pattern i trained without that example. We therefore get–as a bonus of the TAP approach–the following leave-one-out estimator for the generalization error

$$\epsilon_{\mathrm{loo}} = \frac{1}{N} \sum_{i=1}^{N} \Theta(-t_i \langle h_i \rangle_{\setminus i}). \tag{32}$$

If ϵ_{loo} were exact, i.e. equal to the exact loo-estimator $\epsilon_{\mathrm{loo}}^{\mathrm{exact}}$ obtained by N–fold cross validation, this estimator would be (almost) unbiased.[2] In appendix D, we will show that the expression for $\langle h_i \rangle_{\setminus i} = \langle h_i \rangle - \lambda_i t_i \alpha_i$ may also be obtained by treating the perturbation of the TAP equations when the ith example is removed from the training set by a linear response approach. This

[2] $\epsilon_{\mathrm{loo}}^{\mathrm{exact}}$ is an unbiased estimator for the generalization error for a training set of size $N-1$. We may also define an additional loo-estimator from the error probability eq. (14), $\epsilon_{\mathrm{pred}} = \frac{1}{N} \sum_i \Phi\left(-\frac{|\langle h_i \rangle_{\setminus i}|}{\sqrt{\lambda_i}} \right)$. However, this estimator is only unbiased when the model assumptions are correct and will therefore in many cases be misleading. This is also observed in simulations.

gives a self-consistency check of the result for $\langle h_i \rangle_{\setminus i}$ obtained within the TAP approximation. In Appendix D, it is furthermore shown how this technique can be generalized to the derivation of loo–estimators for other algorithms, e.g. the naive mean field algorithm (Section 6) and Support Vector Machines (section 7). Other work on approximate loo-estimators for smoothing splines (Xiang & Wahba, 1996) and neural networks (Larsen & Hansen, 1996) is close in spirit to the linear response approach.

6 Naive Mean Field Theory

There are many different ways to define and derive mean field theories, see e.g. (Parisi, 1988). Perhaps the simplest one is to derive an exact relation between the expectations of the random variables of interest and the expectation of a non-linear function of these variables. For spin systems, such expressions are known as Callen identities (see Parisi's (1988) field theory book on page 26). At the end of the calculation, the expectation is naively exchanged with the non-linearity in order to derive a closed self-consistent approximation for the averages.

For our problem, we start from the definition of α_i, eq. (18) and firstly apply a standard Gaussian transform which introduces the imaginary unit $i = \sqrt{-1}$ (not to be confused with the i appearing as an index)

$$\begin{aligned} \alpha_i &= \frac{t_i}{p(\mathbf{t})} \int d\mathbf{h} p(\mathbf{h}) \frac{\partial}{\partial h_i} p(\mathbf{t}|\mathbf{h}) \\ &= \frac{t_i}{p(\mathbf{t})} \int \frac{d\mathbf{h} d\mathbf{y}}{(2\pi)^N} \exp\left(-\frac{1}{2}\mathbf{y}^T \mathbf{C}\mathbf{y} + i\mathbf{y}^T\mathbf{h}\right) \frac{\partial}{\partial h_i} p(\mathbf{t}|\mathbf{h}) \end{aligned} \tag{33}$$

Secondly, integration by parts is applied

$$\alpha_i = \frac{t_i}{p(\mathbf{t})} \int \frac{d\mathbf{h} d\mathbf{y}}{(2\pi)^N} (-iy_i) \exp\left(-\frac{1}{2}\mathbf{y}^T \mathbf{C}\mathbf{y} + i\mathbf{y}^T\mathbf{h}\right) p(\mathbf{t}|\mathbf{h}) = -it_i \langle y_i \rangle \ .$$

In the last equality the bracket is understood as a formal average over the joint complex measure of the variables $\mathbf{h}$ and $\mathbf{y}$. Next, we separate the integrations over h_i and y_i from the rest of the variables to show that

$$\begin{aligned} \alpha_i &= t_i \left\langle \frac{\int dh_i dy_i \exp\left(-\frac{1}{2}C_{ii}(y_i)^2 + (-iy_i)(\sum_{j\neq i} C_{ij}(-iy_j) - h_i))\right) \frac{\partial p(t_i|h_i)}{\partial h_i}}{\int dh_i dy_i \exp\left(-\frac{1}{2}C_{ii}(y_i)^2 + (-iy_i)(\sum_{j\neq i} C_{ij}(-iy_j) - h_i))\right) p(t_i|h_i)} \right\rangle \\ &= t_i \left\langle \frac{\int dh_i \exp\left(-\frac{(h_i - \sum_{j\neq i} C_{ij}(-iy_j)))^2}{2C_{ii}}\right) \frac{\partial p(t_i|h_i)}{\partial h_i}}{\int dh_i \exp\left(-\frac{(h_i - \sum_{j\neq i} C_{ij}(-iy_j))^2}{2C_{ii}}\right) p(t_i|h_i)} \right\rangle \end{aligned}$$

$$= \frac{1}{\sqrt{C_{ii}}} \left\langle \frac{D\left(\frac{\sum_{j\neq i} C_{ij}(-iy_j)}{\sqrt{C_{ii}}}\right)}{\Phi\left(t_i \frac{\sum_{j\neq i} C_{ij}(-iy_j)}{\sqrt{C_{ii}}}\right)} \right\rangle \tag{34}$$

If we simply neglect the fluctuations of the field $\sum_{j\neq i} C_{ij}(-iy_j)$ and move the expectation through the non-linearities, we get a self-consistent set of equations for $\alpha_i = -it_i\langle y_i\rangle$. We may call this a 'naive' mean field theory.[3] The explicit expression for α_i becomes

$$\alpha_i \approx \frac{1}{\sqrt{C_{ii}}} \frac{D\left(\frac{\sum_{j\neq i} C_{ij} t_j \alpha_j}{\sqrt{C_{ii}}}\right)}{\Phi\left(t_i \frac{\sum_{j\neq i} C_{ij} t_j \alpha_j}{\sqrt{C_{ii}}}\right)} \ . \tag{35}$$

This can be seen as simplified version of the TAP equations, where (cf. eq. (23)) the complicated expression for λ_i, eq. (28) has been replaced with C_{ii}. This mean field theory is therefore of a lower computational complexity since we avoid the matrix inversion needed in the TAP approach to obtain λ_i.

We may obtain the leave-one-out estimator for naive mean field theory from eq. (48), where $\delta\langle h_i\rangle$ is given by eq. (47) and Ω_i, eq. (49) becomes

$$\Omega_i = C_{ii}\left(\frac{1}{t_i\alpha_i\langle h_i\rangle} - 1\right) \ .$$

Again, we observe that the only difference to TAP is that λ_i has been exchanged with C_{ii} (compare Ω_i which corresponds to Λ_i, eq. (29) for TAP).

7 Support Vector Machine Digression

The relation between variational problems in reproducing kernel Hilbert spaces (such as the learning in support vector machines) and Bayesian Gaussian processes is well known and has been pointed out by various authors, see e.g. (Wahba, 1990; Jaakkola & Haussler, 1999).

We give a simple formulation of support vector learning in the realizable case (neglecting the bias) (Boser, Guyon & Vapnik, 1992; Vapnik, 1995; Schölkopf, Burges & Smola, 1998): Based on the decomposition for the activation function $h(\mathbf{x})$, eq. (5), one tries to find weights w_ρ such that $t_i h(\mathbf{x}_i) \geq 1$ for all examples and $\sum_\rho w_\rho^2$ is minimal. This is therefore an optimization problem with no direct probabilistic interpretation. However, this fits into

[3] The same result can be obtained by a saddlepoint approximation of a suitable partition function (Csató *et. al.*, 1999).

the Gaussian process framework by the following limiting procedure: To introduce the margin, we replace $p(t|h) = \Theta(th)$ by the non-normalized 'Likelihood' $\Theta(th-1)$ such that absolute values of activations smaller than 1 are excluded. Second, the prior is rescaled by introducing a parameter β:

$$p(\mathbf{h}) = \sqrt{\frac{\beta}{(2\pi)^N \det \mathbf{C}}} \exp\left(-\frac{\beta}{2}\mathbf{h}^T\mathbf{C}^{-1}\mathbf{h}\right). \tag{36}$$

In the independent variables w_ρ this prior is simply $\propto \exp[-\frac{\beta}{2}\sum_\rho w_\rho^2]$. Hence, in the limit $\beta \to \infty$, the 'posterior' is dominated by the solution of the SV problem.

Interestingly, although the TAP equations give in general only approximations to posterior averages, it is shown in appendix E that they reduce to the *exact* Kuhn–Tucker conditions which determine the embedding strengths $\hat{\alpha}_i$ for SVMs:

$$(\hat{\alpha}_i = 0 \;\text{ and }\; t_i\langle h_i\rangle \geq 1) \quad \text{or} \quad (\hat{\alpha}_i > 0 \;\text{ and }\; t_i\langle h_i\rangle = 1)\;, \tag{37}$$

where $\langle h_i\rangle = \sum_j C_{ij}t_j\hat{\alpha}_j$. The prediction of the support vector machine is given by $t^{\rm SVM}(\mathbf{x}, D_N) = \mathrm{sgn}\langle h(\mathbf{x})\rangle = \mathrm{sgn}\sum_j C(\mathbf{x},\mathbf{x}_j)t_j\hat{\alpha}_j$.

In appendix E, we show that taking the SVM-limit of the loo-estimator eq. (32), a simple approximation for the loo-estimator for SVMs is obtained

$$\epsilon_{\rm loo}^{\rm SVM} \approx \frac{1}{N}\sum_i^{\rm SV}\Theta\left(1 - \frac{\hat{\alpha}_i}{\left[\mathbf{C}_{\rm SV}^{-1}\right]_{ii}}\right)\;, \tag{38}$$

where the sum goes over SVs only and $\mathbf{C}_{\rm SV}$ denotes the covariance matrix for the SV examples. An alternative derivation of eq. (38) and generalizations are possible along the lines of appendix D (Opper & Winther, 1999b). A similar type of loo estimator for SVM (for a different loss function) has previously been derived by Wahba (1998).

8 Simulations

In this section, we study the performance of the TAP mean field, naive mean field and the support vector machine (SVM) algorithm for three publicly available and widely tested data sets: 'Sonar – Mines versus Rocks' (Gorman & Sejnowski, 1988) 'Leptograpsus Crabs' and 'Pima Indians Diabetes' both (Ripley, 1996). For SVM, we used the Adatron algorithm without bias (Anlauf & Biehl, 1989). The input data was pre-processed by linear rescaling such that over the training set each input variable has zero mean and unit variance. We tested several different covariance functions. All of them are

listed in Appendix B. In almost all cases the Gaussian covariance function turned out to give the best performance. We have therefore chosen only to report results for that covariance function.

Hyperparameters. The free parameters of the algorithms are the hyperparameters of the covariance functions (described in Appendix B). These should be determined from either prior knowledge about the problem or in lack of that from the training data.

In the simulations presented here, we have reduced the number of free hyperparameters either by setting all the $\mathbf{w}$-hyperparameters (see Appendix B) to a common value ('Sonar') or used the values found in a previous study ('Crabs' and 'Pima') (Williams & Barber, 1997). The remaining hyperparameters are chosen as to minimize the value of the leave-one-out estimator. For the optimization using the leave-one-out estimator, we simply make a very rough and probably non-optimal optimization by scanning through a range of values. This is possible because we restrict ourselves to at most three free parameters, where in most cases only one is different from zero. It should be noted that the performance was found to be quite robust against changes in the hyperparameters. In the Conclusion, section 9, we briefly discuss how to make the hyperparameter determination automatic and the Bayesian 'evidence' framework as a possible alternative for determining the hyperparameters.

Test of leave-one-out estimator. The generalization error estimator for the different algorithms have been compared with the exact $\epsilon_{\text{loo}}^{\text{exact}}$ leave-one-out cross-validation error. I.e. the error count we get, when going through all training example, keeping in turn one example out for testing and running the mean field algorithm on the rest. The results are shown in tables 1 and 2. Figure 2 shows the values of different error measures for a range of values for a single hyperparameter. The overall conclusion is that the leave-one-out estimators are very precise. Usually they get at most one classification wrong compared to the exact value. An exception is extremely short–ranged covariance functions shown, see figure 2.

8.1 Sonar Data

'Sonar – Mines versus Rocks' is the data set used by Gorman and Sejnowski in their study of the classification of sonar signals using neural networks (Gorman & Sejnowski, 1988). The task is to train a classifier to discriminate between sonar signals bounced off a metal cylinder and those bounced off a rough cylindrical rock. The dataset contains 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under

various conditions and 97 patterns obtained from rocks under similar conditions. Here we have used the same training/test data split as Gorman and Sejnowski.[4]

Results. For all algorithms the number of hyperparameters were strongly reduced by setting $w_l = 1/\sigma^2 d$ for $l = 1, \ldots, d$, where d is the input dimension. It turned out that the performance is quite robust against the choice of σ^2 as it may seen from figure 2. The simulations results are shown in table 1. The comparison is taken from (Gorman & Sejnowski, 1988). One may observe that the performance of the mean field algorithms are slightly better than that of the SVM and the best two-layer network. However, it is doubtful that this difference is significant.

Table 1: Results for Sonar data.

Algorithm	ϵ_{test}	$\epsilon_{\text{loo}}^{\text{exact}}$	ϵ_{loo}
TAP Mean Field	0.077	0.212	0.212
Naive Mean Field	0.077	0.221	0.221
SVM	0.096	0.202	0.202
Simple Perceptron	0.269		
Best Two-Layer (12 Hidden)	0.096		

Figure 2 shows that the test set is significantly 'easier' than the training set. The solution with lowest test error is almost trivial with all $\alpha_i \sim t\langle h_i \rangle \sim$ constant. Exchanging the training and test set, we find e.g. for $\sigma^2 = 0.05$, $\epsilon_{\text{test}} = 0.173$ and $\epsilon_{\text{loo}} = 0.058$. One may thus conclude that this training/test split is very biased. Running the algorithms on other splits gave a non-trivial solution, with a much better correspondence between the test and leave-one-out error and an error rate somewhere between the two extremes.

In figure 3, we give an example of the 'embedding strength' α_i and the 'stability' $t_i\langle h_i \rangle$ found for TAP mean field, naive mean field and SVMs used under exactly the same condition, i.e. same training set and covariance matrix. The results are normalized such that the squared length of the weight vector $\sum_{ij} \alpha_i t_i C_{ij} t_j \alpha_j$ is the same for all three cases. The results show a clear correlation between the solutions found by the three algorithms. However, the mean field algorithms give rise to both smaller and larger stabilities (margins) than found for the SVM. This, however, has no major effect on performance.

[4]May be obtained from `http://www.boltz.cs.cmu.edu/benchmarks/sonar.html`.

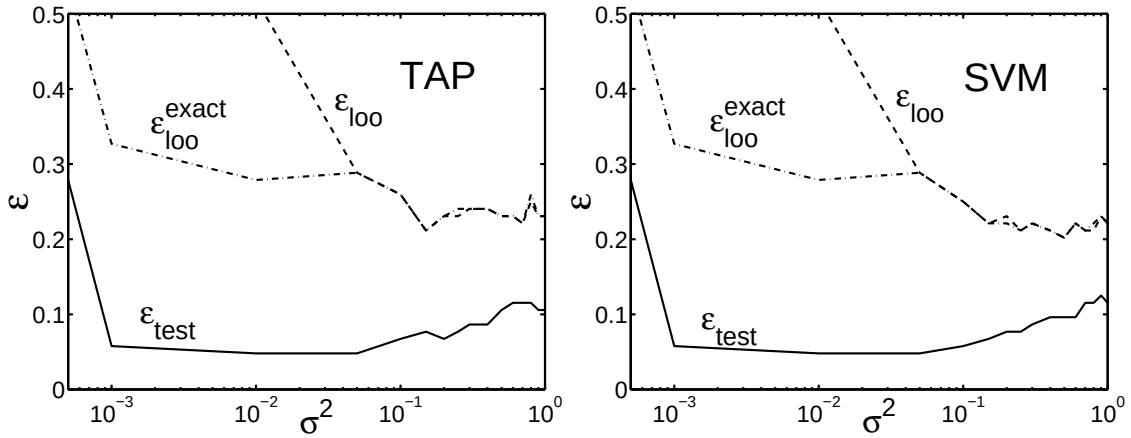

Figure 2: The test error ϵ_{test} (full line) and the leave-one-out estimator ϵ_{loo} (dashed line) and $\epsilon_{\text{loo}}^{\text{exact}}$ (dash-dotted line) as a function of σ^2 for the Sonar data set. The left plot is for TAP and the right plot is for SVM.

8.2 Leptograpsus Crabs and Pima Indians Diabetes

Both the 'Leptograpsus Crabs' and 'Pima Indians Diabetes' data set has been made available by Ripley (1996).[5]

For 'Leptograpsus Crabs' the task is to classify the sex of crabs on the basis of five anatomical attributes and a color attribute. There are 50 examples available for each color and sex, making a total of 200 examples. 20 examples for each color and sex in the data file are used for training (making 80 examples in total) and the rest are used for testing. We use the same training/test split as Williams and Barber (1997).

The 'Pima Indians Diabetes' data set is used as in previous studies with a training/test split of 200 and 332 examples respectively (Ripley, 1996). The task is to diagnose diabetes in a population of women of Pima Indian heritage based upon the following information: number of pregnancies, plasma glucose concentration in an oral glucose tolerance test, diastolic blood pressure, triceps skin fold thickness, serum insulin, body mass index, diabetes pedigree function and age. The baseline error rate, obtained by simply classifying each test input as positive, is 0.33.

Results. The $\mathbf{w}$-hyperparameters were chosen to have values found by Williams & Barber (1997) based on the Laplace approximation to the integral over the Gaussian fields and ML-II estimation for the hyperparameters. The results are given in table 2. The comparisons are taken from Williams & Barber (1997) for 'Laplace GP' and all others are from Ripley (1994,1996). Our results are found to be close to the state of the art. The performance is

[5]May be obtained from `http://www.stats.ox.ac.uk/~ ripley/PRbook/`.

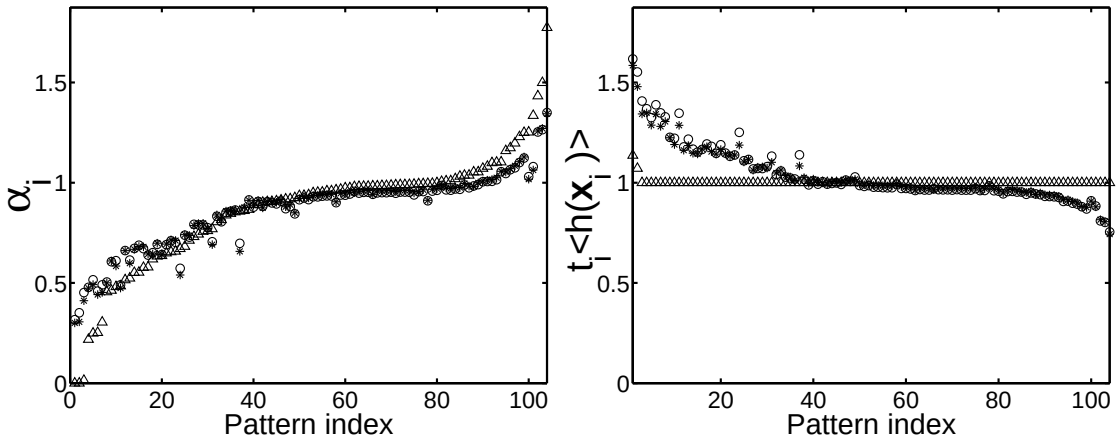

Figure 3: Left figure: The 'embedding strengths' α_i plotted for the different algorithms for the Sonar data. The right figure: The mean activation $t_i\langle h_i\rangle$ for the different algorithms (same ordering as the left plot). The triangles are for support vectors, circles are for naive mean field theory and stars are for TAP mean field theory. They are sorted in ascending order according to their support vector α_i value and the TAP and 'naive' mean field solution is rescaled to the length of the support vector solution.

quite sensitive to the choice of $\mathbf{w}$-hyperparameters, so one may hopefully improve upon these results when the hyperparameters are tuned to the specific algorithm. For 'Pima Indians Diabetes', the difference in performance between algorithms are small and contrary to the other data sets all covariance functions tested have similar performance.

9 Conclusion

In this paper, we have presented a mean field approach to binary classification with Gaussian processes. Our method is an extension of the TAP mean field theory known in Statistical Physics. However, we have avoided making special assumptions about the distribution of the randomness of input data. The derivation of the resulting non-linear equations for posterior means is new and may also be applied to other problems in probabilistic modeling. Our mean field method can be applied to statistical models which are non-smooth functions of their parameters (like noise–free classification), where approaches based on Taylor expansions of the likelihood or the posterior would fail.

A 'naive' mean field algorithm, which can be considered as a simplified version of the TAP algorithm with a lower computational complexity has also been derived. We have further shown how support vector machines (SVMs) can be obtained from a limiting procedure of the TAP approach. The built-

Table 2: Results for 'Leptograpsus Crabs' and 'Pima Indians Diabetes'.

	Leptograpsus Crabs				Pima Indians Diabetes			
Algorithm	ϵ_{test}	$\epsilon_{\text{loo}}^{\text{exact}}$	ϵ_{loo}	Err.	ϵ_{test}	$\epsilon_{\text{loo}}^{\text{exact}}$	ϵ_{loo}	Err.
TAP Mean Field	0.033	0.037	0.037	4	0.190	0.250	0.255	63
Naive Mean Field	0.017	0.025	0.025	2	0.193	0.245	0.245	64
SVM	0.050	0.037	0.037	6	0.199	0.250	0.250	66
Laplace GP–Hybrid Monte Carlo				3				68
–Maximum Likelihood-II				4				69
Best Two–Layer Network				4				75
Simple Perceptron				8				67
Logistic Regression				4				66
MARS (degree=1)				8				75
Projection Pursuit (4 Ridge Functions)				3				75
Gaussian Mixture								64

in leave-one-out estimator for the generalization error provided by the TAP mean field method naturally extend to the SVM and naive mean field case.

Simulation result on three small benchmark data sets 'Sonar – Mines versus Rocks', 'Leptograpsus Crabs' and 'Pima Indians Diabetes' all give promising results matching the state of the art performance. However, there are still a number of unresolved questions that need to be addressed before these new algorithms may be regarded as practical tools. We will discuss these in the following.

Model selection. So far we have not dealt with automatic determination of hyperparameters. After having reduced the number of hyperparameters, the remaining ones have been found by minimizing the leave-one-out estimate over a range of trial values. To be able to work with a higher number of hyperparameters, the search should be automated preferable by a gradient descent method on a suitable differentiable function which rules out the discrete leave-one-out error count.

A well established candidate for such a function which exploits the full power of the Bayesian approach is the total probability of the data given the model. This is termed the *evidence* in Bayesian terminology (Mackay, 1992), or the *stochastic complexity* in Rissanen's MDL approach. The negative log–evidence corresponds to the so called *free energy* in the language of statistical physics for which approximations are also available within mean field theory (Opper & Winther, 1999a, Csató *et. al.*, 1999). Preliminary results of this

approach are very promising and will be reported in a forthcoming paper.

Assessing the validity of the mean field approximation. When the iterative scheme for solving the mean field equations fails, we have so far no way of telling whether the equations have actually no solution or whether our numerical method is simply not appropriate for the specific choice of parameters. A possible way to approach this problem is its conversion into the minimization of a suitable cost function like the TAP free energy in (Thouless, Anderson & Palmer, 1977; Opper & Winther 1999a).

Quality of Mean Field Approximations. The quality of the mean field approximations may be directly assessed by a comparison with Monte Carlo sampling (Neal, 1997), which can approximate posterior averages with arbitrary accuracy when enough computer time is invested. On the other hand, for some toy models with high dimensional input spaces, it should be possible to derive exact Bayesian learning curves by the replica method of statistical mechanics against which our mean field methods can be tested (Opper & Winther, 1996).

Computational complexity. The computational complexity of the TAP mean field algorithm is $\mathcal{O}(N^3)$, where N is number of training examples since it requires the inversion of a $N \times N$ matrix. This makes the algorithm impractical for data sets larger than a few thousands. The complexity of the 'naive' mean field algorithm–on the other hand–should not scale worse than $\mathcal{O}(N^2)$ since one iteration update only requires N inner products of N terms and the iterative process is expected to converge in a finite number of steps. However, its leave-one-out estimate still requires a matrix inversion which is of $\mathcal{O}(N^3)$. If instead the evidence framework is used–which is only $\mathcal{O}(N^2)$ (Csató *et. al.*, 1999) for this algorithm–naive mean field theory could be an interesting alternative to SVMs also for larger datasets.

The similarity between the results of the mean field theory and the SVM algorithm suggests further simplifications. In cases where SVMs lead to sparse representations based on a small number of support vectors, the mean field algorithms are expected to converge also to a small number of large embedding strengths. Setting the remaining ones to zero and leaving the corresponding examples out of the training set at an early stage of the iteration may speed up the algorithm significantly.

Finally, one may think about hybrid approaches which combine the strengths of both the SVM and the Bayesian approaches. One could, e.g. calculate the embedding strengths directly from the SVM algorithm, hereby exploiting the sparseness directly, but use a mean field approximation to the Bayesian

evidence as a yardstick for estimating the hyperparameters in the kernel.

Extension to other data models It would be interesting to try to apply our TAP approach for Gaussian process models to other types of likelihoods. A natural choice would be the problem of classification with more than two classes. A likelihood for this problem can be simply obtained from a *softmax* function, where each class has its own Gaussian process. Our TAP approach requires the integration of such a likelihood over Gaussian distributions, a task which unfortunately can no longer be done exactly. Hence, further approximations for treating such (usually) low-dimensional integrations will be necessary.

Acknowledgments

We are thankful to Pál Ruján, Sara A. Solla and Grace Wahba for discussions and to Chris Williams and two anonymous referees for very constructive comments and suggestions. This research is supported by the Swedish Foundation for Strategic Research.

A Input Noise

In this appendix, we discuss the inclusion of input noise within the probit model and the modification of the mean field algorithms implied. We define the input noise model as the addition of an independent random variables to the activation function in the likelihood eq. (1):

$$p(t|h(\mathbf{x}), \xi(\mathbf{x})) = \Theta(t(h(\mathbf{x}) + \xi(\mathbf{x}))) \ . \tag{39}$$

We will assume that the noise is Gaussian with zero mean and variance v_2, i.e. *probit* regression (Neal, 1997). With Gaussian noise, the original step function likelihood will be modified to a sigmoidal shaped (error) function with the gain given by the variance of the noise

$$p(t|h) = \int d\xi \, p(t|h, \xi) p(\xi) = \Phi\left(\frac{th}{\sqrt{v_2}}\right) .$$

Alternatively, but completely equivalent, one may shift variables to a noisy process $h(\mathbf{x}) + \xi(\mathbf{x})$. This process will also be Gaussian with a modified covariance matrix $C_{ij} + v_2 \delta_{ij}$. The Likelihood (and the mean field equations) for this process is therefore the same as for the noise-free case. The noise is simply included as the extra term in the covariance matrix. In the simulations we will stick to the latter formulation. Importantly, the two formulations are

equivalent for both TAP and 'naive' mean field theory, i.e. the Bayes prediction and the leave-one-out estimate $\langle h_i \rangle_{\setminus i}$ are the same in both formulations.[6] Some the quantities in the theory are not invariant, e.g. $\langle h_i \rangle$ in the original process is equivalent to $\langle h_i \rangle + v_2 t_i \alpha_i$ in the noisy process whereas others e.g. α_i remain unchanged.

B Covariance Functions

A thorough discussion of covariance functions and their properties is given in Abrahamsen (1997). We have tested the following covariance functions

1. **Simple Perceptron:** $C(\mathbf{x}, \mathbf{x}') = \sum_{l=1}^{d} w_l x_l x_l' + v_1$.
2. **Infinite Net:** $C(\mathbf{x}, \mathbf{x}') = v_0 \frac{2}{\pi} \arcsin\left(\frac{\sum_l w_l x_l x_l'}{\sqrt{(1+\sum_l w_l x_l x_l)(1+\sum_l w_l x_l' x_l')}} \right) + v_1$.
3. **Gaussian:** $C(\mathbf{x}, \mathbf{x}') = v_0 \exp\left(-\frac{1}{2} \sum_l w_l (x_l - x_l')^2 \right) + v_1$.
4. **Cauchy:** $C(\mathbf{x}, \mathbf{x}') = v_0 \frac{1}{(1+\sum_l w_l (x_l - x_l')^2)^\delta} + v_1$.

The first two are obtained from feed-forward neural network models which converge to Gaussian processes. In both cases the prior on the weights and thresholds is spherical Gaussian. The infinite net covariance function is obtained from a two-layer network with linear output unit in the limit of infinite number of hidden units with a sigmoidal (error function) activation function given by $g(x) = 2\Phi(x) - 1$ (Neal, 1996; Williams, 1997). The sign-activation function $g(x) = \mathrm{sgn} x$ may also be treated in the infinite net limit and simply amounts to removing the terms of 1 in the denominator of the infinite net covariance function. The Gaussian covariance function is a well known example of an exponential covariance function. It may be obtained from a radial basis network with an infinite number of hidden units (Williams, 1997). The Cauchy or rational quadratic covariance function is defined for all $\delta > 0$ (Abrahamsen, 1997). In the simulations we have simply chosen to work with $\delta = 1$.

Since the properties of the solution is independent of the scale of Gaussian fields we may fix v_0 to say $v_0 = 1$. The length scale for input dimension l is defined by $1/\sqrt{w_l}$. Important inputs will be well-defined and thus be associated with a relative short length scale and vice versa for unimportant length scales. v_1 correpsonds to the variance of the output threshold. To include the effect of Gaussian noise on the fields, a term v_2 is–as discussed in Appendix A–added to the diagonal of the covariance *matrix*.

[6] Note that in a MAP approach (Williams & Barber, 1998) the two formulation are *not* equivalent.

C Second Moments λ and λ_i

In this appendix, we derive mean field expressions for the second moments

$$\begin{aligned} \lambda &= \langle h^2(\mathbf{x})\rangle - \langle h(\mathbf{x})\rangle^2 \\ \lambda_i &= \langle h_i^2\rangle_{\setminus i} - \langle h_i\rangle_{\setminus i}^2 \ . \end{aligned}$$

using a linear response method. We first derive λ in some detail and thereafter indicate how to generalize this result to λ_i. By integrating by parts using eq. (16), we find

$$\langle h^2(\mathbf{x})\rangle = C(\mathbf{x},\mathbf{x}) + \sum_k C(\mathbf{x},\mathbf{x}_k)\frac{1}{p(\mathbf{t})}\int d\mathbf{h}dh\, p(\mathbf{h},h)\, h(\mathbf{x})\frac{\partial}{\partial h_k}p(\mathbf{t}|\mathbf{h}) \ . \quad (40)$$

To rewrite the last term so that we may apply a linear response argument, it should be expressed as a derivative of a known quantity, in this case $\langle h(\mathbf{x})\rangle$. This is achieved by introducing fields $\boldsymbol{\xi} = \xi_1,\ldots,\xi_N$ added to the random activations $\mathbf{h}$ in the Likelihood term, taking the derivative wrt. this field and setting $\boldsymbol{\xi}=0$ afterwards, i.e. $\frac{\partial}{\partial h_k}p(\mathbf{t}|\mathbf{h}) = \frac{\partial}{\partial \xi_k}p(\mathbf{t}|\mathbf{h}+\boldsymbol{\xi})\Big|_{\boldsymbol{\xi}=0}$:

$$\begin{aligned} \lambda &= C(\mathbf{x},\mathbf{x}) + \sum_k C(\mathbf{x}_k,\mathbf{x})\frac{\partial}{\partial\xi_k}\left(\frac{1}{p(\mathbf{t}|\boldsymbol{\xi})}\int d\mathbf{h}dh\, p(\mathbf{h},h)\, h(\mathbf{x})\, p(\mathbf{t}|\mathbf{h}+\boldsymbol{\xi})\right)\Big|_{\boldsymbol{\xi}=0} \\ &= C(\mathbf{x},\mathbf{x}) + \sum_k C(\mathbf{x}_k,\mathbf{x})\frac{\partial\langle h(\mathbf{x})\rangle}{\partial\xi_k}\Big|_{\boldsymbol{\xi}=0} \ . \quad (41) \end{aligned}$$

where $p(\mathbf{t}|\boldsymbol{\xi}) = \int d\mathbf{h}p(\mathbf{h})p(\mathbf{t}|\mathbf{h}+\boldsymbol{\xi})$. Note that moving the derivative to the left of the normalization constant gives rise to an additional factor of $-\langle h(\mathbf{x})\rangle^2$ compared to eq. (40).

To calculate the linear responses $\frac{\partial\langle h(\mathbf{x})\rangle}{\partial\xi_k}$ within the Gaussian approximations of the TAP approach, we make a further approximation and assume that by adding small fields we need only to consider the changes in the means of the fields, but can neglect changes in the variances. Using eq. (17), we get

$$\delta\langle h(\mathbf{x})\rangle = \sum_j C(\mathbf{x},\mathbf{x}_j)t_j\delta\alpha_j. \quad (42)$$

We therefore have to solve for $\delta\alpha_j$ in terms of $\boldsymbol{\xi}$. The shifts of the α_j are related to the shifts in the means of the fields via eq. (19)

$$\delta\langle h_i\rangle = \xi_i + \sum_j C_{ij}t_j\delta\alpha_j \quad (43)$$

where the shift $\delta\alpha_j$ is found from eq. (26):

$$\delta\langle h_i\rangle \approx -\Lambda_i t_i\delta\alpha_i \quad (44)$$

with Λ_i defined by eq. (29). Substituting $\delta\langle h_i\rangle$ into eq. (43) and solving for $\delta\alpha_i$, we finally arrive at eq. (27).

We can repeat the same steps to derive equations for λ_i. The only difference is the fact that all posterior distributions must be replaced by the corresponding posteriors where the ith pattern is absent. Analogous to eq. (27), we find

$$\lambda_i \approx C_{ii} - \sum_{jk\neq i} C_{ij} \left[(\mathbf{\Lambda}^{\setminus i} + \mathbf{C}^{\setminus i})^{-1} \right]_{jk} C_{ki} \ , \tag{45}$$

where $\mathbf{\Lambda}^{\setminus i}$ and $\mathbf{C}^{\setminus i}$ denote the $N-1 \times N-1$ reduced matrices where the ith row and column are deleted. Here, we have assumed that the change in Λ_j is negligible when the ith pattern is removed ($i \neq j$). Using the following identity for the partitioned inverse, see e.g. (Press, Teukolsky, Vetterling & Flannery, 1992)

$$\left[\mathbf{A}^{-1} \right]^{-1}_{ii} = A_{ii} - \sum_{jk\neq i} A_{ij} \left[(\mathbf{A}^{\setminus i})^{-1} \right]_{jk} A_{ki} \ , \tag{46}$$

the expression may be simplified to the final result eq. (28).

D Leave-One-Out from Linear Response

The subsequent derivation of an approximate leave-one-out (loo) estimator is closely in spirit to other similar procedures which rely on first order expansions. These may be motivated by the assumption of small changes in appropriate variables, when an example is removed from the training set. See e.g. the work of Xiang & Wahba (1996) who develop an approximate loo-estimator for smoothing splines. For applications to neural networks, see eg. (Larsen & Hansen, 1996). For simplicity, we will use the equality (=) in all expressions and expect that the reader will understand, which of them are approximations.

Assume we remove example j from the training set and the algorithm is retrained on the remaining ones, then the mean of the fields of the remaining inputs are changed according to eq. (19)

$$\delta\langle h_i\rangle = \sum_{k\neq j} C_{ik} t_k \delta\alpha_k - C_{ij} t_j \alpha_j \ . \tag{47}$$

The general leave-one-out estimator is then simply

$$\epsilon_{\text{loo}} = \frac{1}{N} \sum_{i=1}^{N} \Theta \left(-t_i (\langle h_i\rangle + \delta\langle h_i\rangle) \right) \ . \tag{48}$$

with j set equal to i in eq. (47).

To make the following derivation as general as possible we will assume that the algorithm produces embedding strengths α_i which are found as the solution of an equation $\alpha_i = f(\langle h_i \rangle, \alpha_i)$. Examples are eqs. (23) and (35) for respectively TAP and naive mean field theory. There may be further dependencies on other quantities as well, but we will assume that their response to the change of $\langle h_i \rangle$ or α_i is negligible. The change $\delta\langle h_i \rangle$ can be expressed by $\delta\alpha_i$ assuming that all changes are small within linear response $\delta\alpha_i = \frac{\partial f}{\partial \langle h_i \rangle}\delta\langle h_i \rangle + \frac{\partial f}{\partial \alpha_i}\delta\alpha_i$, where we analogous to eq. (44) get $\delta\langle h_i \rangle = -\Omega_i t_i \delta\alpha_i$ with the definition

$$\Omega_i \equiv t_i \frac{1}{\frac{\partial f}{\partial \langle h_i \rangle}} \left(\frac{\partial f}{\partial \alpha_i} - 1 \right) \ . \tag{49}$$

Hence, we may solve for $\delta\alpha_k$

$$t_k \delta\alpha_k = \sum_{i \neq j} \left(\left[\mathbf{\Omega}^{\setminus j} + \mathbf{C}^{\setminus j} \right]^{-1} \right)_{ki} C_{ij} t_j \alpha_j \ . \tag{50}$$

Finally, in order to find $\delta\langle h_i \rangle$ when example i itself is removed from the training set, we set $j = i$ in eq. (47) with $\delta\alpha_k$ from eq. (50). Using the matrix identity eq. (46), we get

$$\delta\langle h_i \rangle = -\left\{ \frac{1}{[(\mathbf{\Omega} + \mathbf{C})^{-1}]_{ii}} - \Omega_i \right\} t_i \alpha_i \ . \tag{51}$$

To make a self-consistent check for TAP mean field theory, we note that $\delta\langle h_i \rangle$ should be equal to $\langle h_i \rangle_{\setminus i} - \langle h_i \rangle = -\lambda_i t_i \alpha_i$. This is indeed found to be the case: compare with λ eq. (28) where Ω_i eq. (49) is identified with Λ_i eq. (29).

E Support Vector Machines

In this appendix, we consider the SVM limit of the TAP mean field equations. TAP equations for SVM learning in the simple perceptron for spherical input distribution and large input dimensionality have previously been derived by (Wong, 1995). We firstly re-derive the Kuhn–Tucker condition from the TAP equations and secondly show that the leave-one-out (loo) estimator becomes especially simple in this limit.

We can show self-consistently that the limit $\beta \to \infty$ in eq. (36) is equivalent to sending the variances $\lambda_i \to 0$ in the TAP approach. Having the factor β in the prior is the same as rescaling the covariance matrix by a factor of $1/\beta$, thus eq. (19) becomes

$$\langle h_i \rangle = \sum_j C_{ij} t_j \hat{\alpha}_j \ .$$

where we have introduced the rescaled (and finite) embedding strength $\hat{\alpha}_i \equiv \alpha_i/\beta$. Using the asymptotic expression the error function

$$\frac{D(z)}{\Phi(z)} \to -z\Theta(z) \quad \text{for} \quad |z| \to \infty$$

in eq. (23) (with an extra factor minus 1 for the margin), we obtain $\hat{\alpha}_i = \frac{1-t_i\langle h_i\rangle_{\backslash i}}{\beta\lambda_i}\Theta(1 - t_i\langle h_i\rangle_{\backslash i})$. Inserting this back into eq. (26), we get $t_i\langle h_i\rangle = 1 + (t_i\langle h_i\rangle_{\backslash i} - 1)\Theta(t_i\langle h_i\rangle_{\backslash i} - 1)$. Putting the expressions for $\hat{\alpha}_i$ and $t_i\langle h_i\rangle$ together, we arrive at the Kuhn–Tucker condition eq. (37).

For the loo-estimator eq. (32), we need to compute the cavity average of the activation $\langle h_i\rangle_{\backslash i}$. When $t_i\langle h_i\rangle_{\backslash i} > 1$, we see immediately that $t_i\langle h_i\rangle = t_i\langle h_i\rangle_{\backslash i} > 1$ so non-support vectors do not contribute to the loo error. We need only to consider support vectors for which $t_i\langle h_i\rangle = 1$ and thus $t_i\langle h_i\rangle_{\backslash i} = 1 - \lambda_i\beta\hat{\alpha}_i$. We therefore have to determine $\lambda_i\beta$ which is done by taking the $\lambda_i \to 0$ limit in the expressions for Λ_i eq. (29) and thereafter in the expression for λ_i eq. (28) giving

$$\lambda_i = \begin{cases} \frac{1}{\beta\left[\mathbf{C}_{\mathrm{SV}}^{-1}\right]_{ii}} & \text{for} \quad t_i\langle h_i\rangle_{\backslash i} < 1 \\ \frac{C_{ii}}{\beta} & \text{for} \quad t_i\langle h_i\rangle_{\backslash i} > 1 \ , \end{cases}$$

where $\mathbf{C}_{\mathrm{SV}}$ denotes the covariance matrix for the support vector examples. The result for λ_i proves our self-consistent assumption: $\lambda_i \to 0$ for $\beta \to \infty$. Finally, inserting the result for $t_i\langle h_i\rangle_{\backslash i}$ in eq. (32) gives the loo-estimator for SVMs eq. (38).

References

P. Abrahamsen, A Review of Gaussian Random Fields and Correlation Functions, Technical report 917, Norwegian Computing Center, Oslo, Norway. 2nd edition (1997).

J. K. Anlauf and M. Biehl, The AdaTron: An Adaptive Perceptron Algorithm, Europhys. Lett. **10**, 687 (1989).

D. Barber and C. K. I. Williams, Gaussian Processes for Bayesian Classification via Hybrid Monte Carlo, in *Neural Information Processing Systems 9*, M . C. Mozer, M. I. Jordan and T. Petsche, eds., 340-346. MIT Press (1997).

J. O. Berger, *Statistical Decision theory and Bayesian Analysis*, Springer, New York, 1985.

B. Boser, I. Guyon, and V. N. Vapnik, A training algorithm for optimal margin classifiers, 5th Annual Workshop on Computational Learning Theory, Pittsburgh ACM, 144-152 (1992).

C. Cortes and V. N. Vapnik, Support Vector Machines, Machine Learning **20**,1-25 (1995).

L. Csató, E. Fokoué, M. Opper, B. Schottky and Ole Winther, Efficient Approaches to Gaussian Process Classification, Submitted to *NIPS'99* (1999).
M. N. Gibbs and D. J. C. Mackay, Variational Gaussian Process Classifiers, Preprint Cambridge University (1997).

R. P. Gorman and T, J. Sejnowski, Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets, Neural Networks **1**, 75 (1988).

T. S. Jaakkola and D. Haussler, Probabilistic Kernel Regression Models, to appear in: Proceedings of the 1999 conference on AI and Statistics.

M. I. Jordan ed., *Learning in Graphical Models*, MIT Press (1999).

J. Larsen and L. K. Hansen, Linear Unlearning for Cross-Validation, Advances in Computational Mathematics, **5**, 269-280 (1996).

D. J. C. Mackay, Bayesian Interpolation, Neural Computation **4** 415-447 (1992).

D. J. C. Mackay, Gaussian Processes, A Replacement for Neural Networks, NIPS tutorial 1997, may be obtained from
`http://wol.ra.phy.cam.ac.uk/pub/mackay/`.

M. Mézard and G. Parisi and M. A. Virasoro, *Spin Glass Theory and Beyond*, Lecture Notes in Physics, 9, World Scientific (1987).

R. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics, Springer (1996).

R. M. Neal, Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification, Technical Report 9702, Dept. of Statistics, University of Toronto (1997).

M. Opper and O. Winther, A Mean Field Approach to Bayes Learning in

Feed-Forward Neural Networks, Phys. Rev. Lett. **76**, 1964 (1996).

M. Opper and O. Winther, A Mean Field Algorithm for Bayes Learning in Large Feed-Forward Neural Networks, in *Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan and T. Petsche, eds., 225-231. MIT Press (1997).

M. Opper and O. Winther, Mean Field Methods for Classification with Gaussian Processes, in *Advances in Neural Information Processing Systems 11 (NIPS'98)*, M. S. Kearns, S. A. Solla, and D. A. Cohn, eds., MIT Press, Cambridge, MA, (1999a).

M. Opper and O. Winther, Gaussian Processes and SVM: Mean Field Results and Leave-One-Out, in *Large Margin Classifiers*, P. Bartlett, B. Schölkopf, D. Schuurmans and A. Smola eds., MIT Press (1999b).

A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw - Hill Series in Electrical Engineering, 2nd edition (1984).

G. Parisi, *Statistical field theory*, Frontiers in Physics, Addison-Wesley (1988).

G. Parisi and M. Potters, Mean-Field Equations for Spin Models with Orthogonal Interaction Matrices, J. Phys. A: Math. Gen. **28** 5267 (1995).

T. Plefka, Convergence Condition of the TAP Equation for the Infinite-Range Ising Spin Glass, J. Phys. A **15**, 1971 (1982).

W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C, Cambridge University Press (1992).

B. D. Ripley, Flexible Non-Linear Approaches to Classification, in *From Statistics to Neural Networks*, V. Cherkassy, J, H. Friedman and H. Wechsler eds., 105, Springer (1994).

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

B. Schölkopf, C. J. C. Burges and A. J. Smola, editors, Advances in Kernel Methods: Support Vector Learning, The MIT Press (1998).

D. J. Thouless, P. W. Anderson and R. G. Palmer, Solution of a 'Solvable Model of a Spin Glass', Phil. Mag. **35**, 593 (1977).

V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag New York (1995).

C. K. I. Williams, Computing with Infinite Networks, in *Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan and T. Petsche, eds., 295-301. MIT Press (1997).

C. K. I. Williams and D. Barber, Bayesian Classification with Gaussian Processes, IEEE Trans Pattern Analysis and Machine Intelligence , **20** 1342-1351, (1998).

C. K. I. Williams and C. E. Rasmussen, Gaussian Processes for Regression, in *Neural Information Processing Systems 8*, D. S. Touretzky, M. C. Mozer and M. E. Hasselmo eds., 514-520, MIT Press (1996).

G. Wahba, Spline models for Observational Data, CBMS-NSF Regional Conference Series in Applied Mathematics **59**, SIAM, Philadelphia (1990).

G. Wahba, Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV, Technical report no. 984rr, Dept. of Statistics, U. Wisconsin, also published in (Schölkopf, Burges & Smola, 1998).

K. Y. M. Wong, Microscopic Equations and Stability Conditions in Optimal Neural Networks, Europhys. Lett. **30**, 245 (1995).

D. Xiang and G. Wahba, A Generalized Approximate Cross Validation for Smoothing Splines with non-Gaussian Data, Statistica Sinica **6**, 675-692 (1996).