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

February 16, 1999

Abstract

We derive a mean field algorithm for binary classification with Gaussian processes which are 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 nonlinear 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 obvious 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 used 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 random 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.[1] The TAP approach is a controlled approximation in the sense that it becomes exact for simple distributions of the disorder (the 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 *ansatz* for the so called *cavity fields* (Mézard, Parisi & Virasoro, 1987)–which we believe to be reasonably good for a large class of

[1] An application of such a simpler mean field method will be given elsewhere (Opper, Fokue & Winther, 1999).

distributions–from which a closed set of nonlinear 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. It is summarized in Appendix B. In Section 4, a leave-one-out estimator for generalization error of the TAP mean field algorithm is proposed. In Section 5, a 'naive' mean field algorithm is derived using the Callen identity. Support vector machines (SVMs) are obtained from the TAP mean field theory by a maximum a posteriori (MAP) limit in section 6. In section 4 we derive leave-one-out estimators for the generalization error of naive mean field theory and SVMs using a linear a response approach. In section 7, 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 8. Appendices A, C and D discuss respectively the noise model used, the recipe for solving the mean field equations and the covariance functions used.

2 Gaussian Process Models

We consider the following supervised learning problem: A training set, $D_m = \{(\mathbf{x}_i, t_i)|i = 1, \ldots, m\}$ 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 (or likelihood) 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 $t\,h(\mathbf{x})$. However, as will be 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 of the prior over $h(\mathbf{x})$.

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 $\mathbf{h} = (h(\mathbf{x}_1), \ldots, h(\mathbf{x}_m))$ of random variables have a joint Gaussian distribution

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

$\mathbf{m}$ 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 D.

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)

$$h(\mathbf{x}) = \sum_\rho w_\rho \sqrt{\lambda_\rho} \Psi_\rho(\mathbf{x}) \qquad (3)$$

where $\Psi_\rho(\mathbf{x})$ are eigenfunctions of the covariance function (kernel) C with eigenvalues λ_ρ. From eq. (3), we easily get the 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 discrete a set of training data inputs $\mathbf{x}_i$, one only needs the $m+1$ dimensional Gaussian prior distribution $p(\mathbf{h}, h)$ where $\mathbf{h} = h(\mathbf{x}_1), \ldots, h(\mathbf{x}_m)$ 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

$$p(\mathbf{h}, h|\mathbf{t}) = \frac{1}{Z}\prod_{i=1}^{m} p(t_i|h(\mathbf{x}_i))\; p(\mathbf{h}, h) = \frac{1}{Z} p(\mathbf{t}|\mathbf{h})\; p(\mathbf{h}, h)\;, \tag{4}$$

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

$$\begin{aligned} Z &= \int d\mathbf{h}dh\; p(\mathbf{t}|\mathbf{h})p(\mathbf{h}, h) \\ &= \int d\mathbf{h}\; p(\mathbf{t}|\mathbf{h})p(\mathbf{h}) \end{aligned} \tag{5}$$

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

$$p(h|\mathbf{t}) = \int d\mathbf{h}\; p(\mathbf{h}, h|\mathbf{t})\;. \tag{6}$$

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

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

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

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

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

$$\begin{aligned} t^{\mathrm{Bayes}}(D_m, \mathbf{x}) &= \mathrm{sgn}\left[p(+1|\mathbf{t}) - p(-1|\mathbf{t})\right] \\ &= \mathrm{sgn}\langle \mathrm{sgn} h\rangle\;. \end{aligned} \tag{8}$$

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 $p(-t^{\mathrm{Bayes}}(D_m, \mathbf{x})|\mathbf{t})$.

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 (Opper, Fokue & Winther, 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

$$p(h(\mathbf{x})|\mathbf{t}) = \frac{1}{\sqrt{2\pi\lambda}} \exp\left(-\frac{(h - \langle h \rangle)^2}{2\lambda}\right) \tag{9}$$

To motivate such an approximation, we may look at the expansion eq. (3). $h(\mathbf{x})$ is the projection of a high dimensional (weight) random vector with components w_ρ onto a random direction given by the components $\sqrt{\lambda_\rho}\Psi_\rho(\mathbf{x})$ and we may expect, assuming that the 'features' Ψ_ρ are sufficiently weakly dependent, that for *most* directions Ψ, a Gaussian approximation for this projection is reasonable. Note, that a similar reasoning is used in the theory of the well known *projection pursuit* algorithm (Friedman & Tukey, 1974). In the theory of disordered media, a similar approximation is 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. A detailed discussion of the validity of this 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 this Gaussian approximation can be calculated in a self-consistent way. Before doing so, we use the Gaussian approximation to calculate the predictive probability eq. (7)

$$p(t|\mathbf{t}) = \Phi\left(t\frac{\langle h(\mathbf{x}) \rangle}{\sqrt{\lambda}}\right) \ , \tag{10}$$

where Φ is the error function

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

The Bayes classifier becomes

$$t^{\text{Bayes}}(\mathbf{x}, D_m) = \text{sgn}\langle h(\mathbf{x})\rangle \ . \tag{11}$$

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

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

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

$$p(h(\mathbf{x})|\mathbf{t}, t) = \frac{p(t|h(\mathbf{x}))p(h(\mathbf{x})|\mathbf{t})}{\int dh \ p(t|h)p(h|\mathbf{t})} \ . \tag{13}$$

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 $p(h(\mathbf{x})|\mathbf{t}, t)$.

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). 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 $3N$ 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 ansatz 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 &

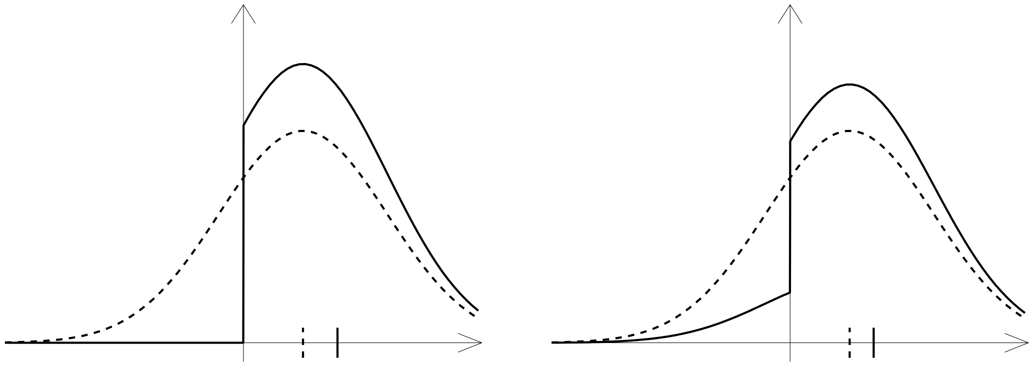

Figure 1: The predictive posterior of $h(\mathbf{x})$, $p(h(\mathbf{x})|\mathbf{t})$ (dashed line) and the posterior $p(h(\mathbf{x})|\mathbf{t},t)$ (solid line) after the addition of the $N+1$th example $(\mathbf{x},t)$. In the figure $t=1$. The left plot is for noise-free case and the right is for output noise (see Appendix A). The lines on the x-axis indicates the mean value of the two distributions.

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 $\langle h(\mathbf{x}_i)\rangle$. Our approximative treatment will be mainly based on exact relations for posterior mean

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

and posterior variance, which can be derived using integration by parts and the relation

$$hp(\mathbf{h},h) = -\left[\sum_i C(\mathbf{x},\mathbf{x}_j)\frac{\partial}{\partial h_j} + C(\mathbf{x},\mathbf{x})\frac{\partial}{\partial h}\right] p(\mathbf{h},h)\ . \tag{14}$$

This enables us to show that for any point $\mathbf{x}$

$$\langle h(\mathbf{x})\rangle = \sum_j C(\mathbf{x},\mathbf{x}_j)t_j\alpha_j\ . \tag{15}$$

with

$$\begin{aligned}\alpha_i &= \frac{t_i}{Z}\int d\mathbf{h}p(\mathbf{h})\frac{\partial}{\partial h_i}\prod_j p(t_j|h_j)\\ &= t_i\frac{\langle \frac{\partial}{\partial h_i}p(t_i|h_i)\rangle_{\backslash i}}{\langle p(t_i|h_i)\rangle_{\backslash i}}\ .\end{aligned} \tag{16}$$

Here

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

denotes an average over the *predictive posterior*

$$p(h_i|\mathbf{t}\backslash t_i) = \frac{\int \prod_{j\neq i} dh_j \prod_{j\neq i} p(t_j|h_j)p(\mathbf{h})}{\int d\mathbf{h} \prod_{j\neq i} p(t_j|h_j)p(\mathbf{h})} \ . \tag{17}$$

where $h_i \equiv h(\mathbf{x}_i)$ and $\mathbf{t}\backslash t_i = t_i, \ldots, t_{i-1}, t_{i+1}, \ldots, t_N$ denotes a training set without the ith example. These exact expressions are very interesting, because they allow to express the expected activation $\langle h(\mathbf{x})\rangle$ as a weighted average over the examples of the training set, similar as in the support vector learning approach (Vapnik, 1995). 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 is usually decreasing with distance and 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 we will call 'embedding strength' in the following), which as it may be seen from eq. (16) is always non-negative when $p(t|h)$ is an increasing function of ht.

Finally, introducing the notation $C_{ij} = C(\mathbf{x}_i, \mathbf{x}_j)$ and applying (15) to the the training data points $\mathbf{x} = \mathbf{x}_i$ we have

$$\langle h_i \rangle = \sum_j C_{ij} t_j \alpha_j \tag{18}$$

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. (9)

$$p(h_i|\mathbf{t}\backslash t_i) = \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left(-\frac{(h_i - \langle h_i\rangle_{\backslash i})^2}{2\lambda_i}\right) \tag{19}$$

with variance $\lambda_i = \langle h_i^2\rangle_{\backslash i} - \langle h_i\rangle_{\backslash i}^2$ in order to derive the explicit expression for α_i from eq. (16)

$$\alpha_i = \frac{1}{\sqrt{\lambda_i}} \frac{D\left(\frac{\langle h_i\rangle_{\backslash i}}{\sqrt{\lambda_i}}\right)}{\Phi\left(t_i \frac{\langle h_i\rangle_{\backslash i}}{\sqrt{\lambda_i}}\right)} \tag{20}$$

where we have introduced the Gaussian measure

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

So far we have only 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 moment, we may exploit the following exact relation between the average over the posterior and the predictive posterior

$$\langle \ldots \rangle = \frac{\langle \ldots p(t_i|h_i)\rangle_{\setminus i}}{\langle p(t_i|h_i)\rangle_{\setminus i}} \ . \tag{21}$$

Using integration by parts and the Gaussianity of the predictive posterior, one finds

$$\langle h_i \rangle = \langle h_i \rangle_{\setminus i} + \lambda_i t_i \alpha_i \ . \tag{22}$$

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. Thus within our mean field approximation, we may express the predictive posterior average of the field $\langle h_i \rangle_{\setminus i}$ by other quantities. Note from eqs. (20) and (22) that the equation for α_i is non-linear and thus have to be solved numerically. In sec. 4, 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.

Calculation of λ and λ_i With respect to our Gaussian approximation, it is also necessary to derive exact expressions for the second moments: $\lambda = \langle h^2(\mathbf{x})\rangle - \langle h(\mathbf{x})\rangle^2$ and $\lambda_i \equiv \langle h_i^2 \rangle_{\setminus i} - \langle h_i \rangle_{\setminus i}^2$. These second moments may be found from integration by parts in the same fashion as for eq. (14). We get

$$\langle h^2(\mathbf{x})\rangle = C(\mathbf{x},\mathbf{x}) + \sum_k \frac{C(\mathbf{x}_k,\mathbf{x})}{Z} \int d\mathbf{h} p(\mathbf{h}) \ h(\mathbf{x}) \frac{\partial}{\partial h_k} \prod_l p(t_l|h_l). \tag{23}$$

Hence

$$\begin{aligned}
\lambda &= \langle h^2(\mathbf{x})\rangle - \langle h(\mathbf{x})\rangle^2 = C(\mathbf{x},\mathbf{x}) + \\
&+ \sum_k C(\mathbf{x}_k,\mathbf{x}) \frac{\partial}{\partial \xi_k} \left(\frac{1}{Z} \int d\mathbf{h} p(\mathbf{h}) \ h(\mathbf{x}) \prod_l p(t_l|h_l + \xi_l) \right) \Bigg|_{\xi=0} \\
&= C(\mathbf{x},\mathbf{x}) + \sum_k \frac{\partial \langle h(\mathbf{x})\rangle}{\partial \xi_k} C(\mathbf{x}_k,\mathbf{x})
\end{aligned} \tag{24}$$

In the last two expressions, we have assumed that nonrandom fields ξ_l are added to the random activations h_l. To calculate the linear responses within the Gaussian ansatz, we make one more 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. (15), we get

$$\delta \langle h(\mathbf{x})\rangle = \sum_j C(\mathbf{x},\mathbf{x}_j) t_j \delta\alpha_j. \tag{25}$$

The shifts of the α_j are related to the shifts in the means of the fields via eq. (18) and give

$$\delta \langle h_i \rangle = \xi_i + \sum_j C_{ij} t_j \delta\alpha_j \tag{26}$$

where the shift $\delta\alpha_j$ is found from eq. (22) and results in

$$\delta\langle h_i\rangle = -\Lambda_i t_i \delta\alpha_i \tag{27}$$

with

$$\Lambda_i \equiv -\lambda_i - t_i \left(\frac{\partial \alpha_i}{\partial \langle h_i\rangle_{\setminus i}}\right)^{-1} \tag{28}$$

Inserting into (26) and solving for $\delta\alpha_i$, we finally arrive at

$$\langle h^2(\mathbf{x})\rangle - \langle h(\mathbf{x})\rangle^2 = C(\mathbf{x},\mathbf{x}) - \sum_{jk} C(\mathbf{x},\mathbf{x}_j)\left[(\mathbf{\Lambda}+\mathbf{C})^{-1}\right]_{jk} C(\mathbf{x}_k,\mathbf{x}) \tag{29}$$

An explicit expression for $\frac{\partial \alpha_i}{\partial \langle h_i\rangle_{\setminus i}}$ is with some algebra obtained from eq. (20)

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

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. (29) we find

$$\lambda_i = C_{ii} - \sum_{jk\neq i} C_{ij}\left[(\mathbf{\Lambda}^{\setminus i}+\mathbf{C}^{\setminus i})^{-1}\right]_{jk} C_{ki} \ , \tag{31}$$

where $\mathbf{\Lambda}^{\setminus i}$ and $\mathbf{C}^{\setminus i}$ denote the $m-1\times m-1$ reduced matrices where the i th row and column are deleted. Here, we have assumed that the change in Λ_j is negliable 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)

$$\left[\mathbf{A}^{-1}\right]_{ii}^{-1} = A_{ij} - \sum_{jk\neq i} A_{ij}\left[(\mathbf{A}^{\setminus i})^{-1}\right]_{jk} A_{ki} \tag{32}$$

the expression may be simplified to

$$\lambda_i = \frac{1}{\left[(\mathbf{\Lambda}+\mathbf{C})^{-1}\right]_{ii}} - \Lambda_i \ . \tag{33}$$

We have now obtained a set of $3N$ mean field equations for α_i, λ_i and Λ_i. The mean field equations together with the other relevant quantities of the TAP mean field algorithm are summarized in Appendix B.

4 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 estimator (loo) for the generalization error of the algorithm since

$\text{sgn}\langle h_i\rangle_{\setminus i}$ 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

$$\epsilon_{\text{loo}} = \frac{1}{N}\sum_{i=1}^{N}\Theta(-t_i\,\langle h_i\rangle_{\setminus i}\,) \tag{34}$$

Given that ϵ_{loo} is exact, i.e. equal to the exact loo-estimator $\epsilon_{\text{loo}}^{\text{exact}}$ obtained by N-fold cross validation, this estimator is (almost) unbiased.[2] In the following, 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 i th example is removed from the training set by a linear response approach. This gives a self-consistency check of the result for $\langle h_i\rangle_{\setminus i}$ obtained within the TAP approximation. The same technique can be applied to calculate leave-one-out estimators for other algorithms. This will be used below to derive leave-one-out estimators for generalization error of the naive mean field algorithm (Section 5) and for Support Vector Machines (section 6).

4.1 Linear Response Derivation

The subsequent derivation of an approximate leave-one-out 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 e.g. (Larsen & Hansen, 1996).

To make the following derivation as general as possible we will assume that the algorithm produces embedding strenghts α_i which are found as the solution of an equation $\alpha_i = f(\langle h_i\rangle, \alpha_i)$. There maybe 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 negliable. Assume we remove example i from the training set and the algorithm is retrained on the remaining ones, then the fields of the remaining inputs are changed according to eq. (18)

$$\delta\langle h_j\rangle = \sum_{k\neq i} C_{jk} t_k \delta\alpha_k - C_{ji} t_i \alpha_i \tag{35}$$

the change $\delta\langle h_j\rangle$ can be expressed by $\delta\alpha_j$ assuming that all changes are small

[2] $\epsilon_{\text{loo}}^{\text{exact}}$ is an unbiased estimator for the generalization error for a training set of size $N-1$. We may also define an additional leave-one-out estimator from the error probability eq. (12), $\epsilon_{\text{pred}} = \frac{1}{N}\sum_i \Phi\left(-\frac{|\langle h_i\rangle_{\setminus i}|}{\sqrt{\lambda_i}}\right)$. However, this estimator is only unbiased when the model assumptions are correct and will therefore in many cases be misleading.

within linear response

$$\delta\alpha_j = \frac{\partial f}{\partial \langle h_j \rangle} \delta \langle h_j \rangle + \frac{\partial f}{\partial \alpha_j} \delta\alpha_j \tag{36}$$

or

$$\delta\langle h_j \rangle = -\Omega_j t_j \delta\alpha_j \tag{37}$$

with

$$\Omega_j = t_j \frac{\frac{\partial f}{\partial \alpha_j} - 1}{\frac{\partial f}{\partial \langle h_j \rangle}} \; .$$

Hence, we obtain after some simple steps that

$$t_j \delta\alpha_j = \sum_{k \neq j} \left(\left[\mathbf{\Omega}^{\backslash i} + \mathbf{C}^{\backslash i} \right]^{-1} \right)_{jk} C_{ki} t_i \alpha_i \tag{38}$$

Finally, in order to make the prediction on point i, we have to calculate the total change on the field $\langle h_i \rangle$ from eq. (37)

$$\begin{aligned} \delta\langle h_i \rangle &= \sum_{k \neq i} C_{ik} t_k \delta\alpha_k - C_{ii} t_i \alpha_i \\ &= \left\{ \frac{1}{\left[(\mathbf{\Omega} + \mathbf{C})^{-1} \right]_{ii}} - \Omega_i \right\} t_i \alpha_i \; . \end{aligned} \tag{39}$$

For TAP mean field theory, Ω_i equals Λ_i (compare eqs. (27) and (37)) and from eq. (33), we identify

$$\delta\langle h_i \rangle = -\lambda_i t_i \alpha_i \tag{40}$$

Hence, after retraining, the new field is

$$h_i' = \langle h_i \rangle + \delta\langle h_i \rangle = \langle h_i \rangle_{\backslash i} \; . \tag{41}$$

We have thus accomplished a self-consistent check of the leave-one-out estimator for TAP mean field theory. However, the result for $\delta\langle h_i \rangle$, eq. (39) has a more general applicability as we shall see in the following.

5 Naive Mean Field Theory

There are many different ways to define 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 nonlinearity in order to derive a closed self-consistent approximation for the averages.

For our problem, we need

$$\begin{aligned}
\alpha_i &= \frac{t_i}{Z}\int d\mathbf{h}p(\mathbf{h})\frac{\partial p(t_i|h_i)}{\partial h_i}\prod_{j\neq k}p(t_k|h_k) \\
&= \frac{t_i}{Z}\int d\mathbf{h}d\mathbf{y}\exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{C}\mathbf{y}+i\mathbf{y}^T\mathbf{h}\right)\frac{\partial p(t_i|h_i)}{\partial h_i}\prod_{j\neq i}p(t_j|h_j) \qquad (42)\\
&= \frac{t_i}{Z}\int d\mathbf{h}d\mathbf{y}(-iy_i)\exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{C}\mathbf{y}+i\mathbf{y}^T\mathbf{h}\right)\prod_{j}p(t_j|h_j) = -it_i\langle iy_i\rangle
\end{aligned}$$

In the first equality we applied a standard Gaussian transformation and in the second one an integration by parts. 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

$$\begin{aligned}
\alpha_i &= t_i\left\langle\frac{\int dh_i dy_i\exp\left(-\frac{1}{2}C_{ii}(y_i)^2+(-iy_i)(\sum_{j\neq i}C_{ij}(-iy_j)-h_i))\right)\frac{\partial p(t_i|h_i)}{\partial h_i}}{\int dh_i dy_i\exp\left(-\frac{1}{2}C_{ii}(y_i)^2+(-iy_i)(\sum_{j\neq i}C_{ij}(-iy_j)-h_i))\right)p(t_i|h_i)}\right\rangle \\
&= t_i\left\langle\frac{\int dh_i\exp\left(-\frac{(h_i-\sum_{j\neq i}C_{ij}(-iy_j)))^2}{2C_{ii}}\right)\frac{\partial p(t_i|h_i)}{\partial h_i}}{\int dh_i\exp\left(-\frac{(h_i-\sum_{j\neq i}C_{ij}(-iy_j))^2}{2C_{ii}}\right)p(t_i|h_i)}\right\rangle \\
&= \left\langle\frac{D\left(\frac{\sum_{j\neq i}C_{ij}(-iy_j)}{C_{ii}}\right)}{\Phi\left(t_i\frac{\sum_{j\neq i}C_{ij}(-iy_j)}{C_{ii}}\right)}\right\rangle \qquad (43)
\end{aligned}$$

If we simply neglect the fluctuations of the field $\sum_{j\neq i}C_{ij}(-iy_j)$ and move the expectation through the nonlinearities, 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

$$\alpha_i = \frac{D\left(\frac{\sum_{j\neq i}C_{ij}t_j\alpha_j}{C_{ii}}\right)}{\Phi\left(t_i\frac{\sum_{j\neq i}C_{ij}t_j\alpha_j}{C_{ii}}\right)}\,. \qquad (44)$$

This can be seen as simplified version of the TAP equations, where (cf. eq. (20)) the complicated expression for λ_i, eq. (33) 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, eq. (33).

[3]The same result can–which will be shown elsewhere–be obtained by a saddlepoint approximation of a suitable partition function.

We may obtain the leave-one-out estimator for naive mean field theory from

$$\epsilon_{\mathrm{loo}} = \frac{1}{N} \sum_i \Theta\left(-t_i(\langle h_i \rangle + \delta\langle h_i \rangle)\right) \ ,$$

where $\delta\langle h_i \rangle$ is given by eq. (37) and Ω_i becomes

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

Again, we observe that the only change compared with TAP is that λ_i has been exchanged with C_{ii}.

6 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; Jaakola & Haussler, 1999).

We give a simple formulation of support vector learning in the realizable case (neglecting the bias) (Boser, Guyon & Vapnik, 1992; Vapnik, 1995; Schoelkopf, Burges & Smola, 1998) based on the decomposition eq. (3). Modelling the activation function as

$$h(\mathbf{x}) = \sum_\rho w_\rho \sqrt{\lambda_\rho} \Psi_\rho(\mathbf{x}) \ , \tag{45}$$

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. The prediction of the support vector machine is given by

$$t^{\mathrm{SVM}}(\mathbf{x}, D_m) = \mathrm{sgn} h(\mathbf{x}) \ . \tag{46}$$

This fits into the Gaussian process framework by the following limiting procedure. To fix the lengthscale of h, we replace $p(t|h) = \Theta(th)$ by the function $\Theta(th-1)$ such that absolute values of activations smaller than 1 are excluded. Second, the prior is rescaled by introducing a parameter β:

$$p(\mathbf{h}) = \sqrt{\frac{\beta}{(2\pi)^m \det \mathbf{C}}} \exp\left(-\frac{\beta}{2} \mathbf{h}^T \mathbf{C}^{-1} \mathbf{h} \right) . \tag{47}$$

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. As we will show in a self-consistent manner below, that the limit $\beta \to \infty$ in eq. (47) 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 $1/\beta$. The expression eq. (18) for the mean weights becomes

$$\langle h_i \rangle = \sum_j C_{ij} t_j \alpha_j / \beta$$

It will therefore be convenient to introduce the rescaled (and finite) embedding strength $\hat{\alpha}_i \equiv \alpha_i/\beta$.

Although the TAP equations give in general only approximations to posterior averages, we will show now, that in the limit they reduce to the *exact* Kuhn–Tucker conditions for SVMs. Using the asymptotic expression the the error function

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

in eq. (56), we obtain

$$\hat{\alpha}_i \to \begin{cases} \frac{1 - t_i \langle h_i \rangle_{\setminus i}}{\beta \lambda_i} & \text{for} \quad t_i \langle h_i \rangle_{\setminus i} < 1 \\ 0 & \text{for} \quad t_i \langle h_i \rangle_{\setminus i} > 1 \end{cases} \tag{48}$$

Inserting back into eq. (57), we get

$$t_i \langle h_i \rangle = \begin{cases} 1 & \text{for} \quad t_i \langle h_i \rangle_{\setminus i} < 1 \\ t_i \langle h_i \rangle_{\setminus i} & \text{for} \quad t_i \langle h_i \rangle_{\setminus i} > 1 \end{cases} \tag{49}$$

showing that in the limit, we get the Kuhn-Tucker condition

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

which determines $\hat{\alpha}_i$ in the SV approach. The examples in the second bracket are the so-called support vectors.

We will next derive the corresponding limit of the mean field approximation for the leave-one-out estimator eq. (34). In the SVM limit of TAP, the cavity average of the activation is $t_i \langle h_i \rangle_{\setminus i} = 1 - \lambda_i \beta \hat{\alpha}_i$. 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. (28) and thereafter in the expression for λ_i eq. (33):

$$\Lambda_i \to \begin{cases} 0 & \text{for} \quad t_i \langle h_i \rangle_{\setminus i} < 1 \\ \infty & \text{for} \quad t_i \langle h_i \rangle_{\setminus i} > 1 \end{cases}$$

and

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

where $\mathbf{C}_{\mathrm{SV}}$ denotes the covariance matrix for the support vector examples. Hence, we get a simple approximation for the leave-one-estimator of support vector machines

$$\epsilon_{\mathrm{loo}}^{\mathrm{SVM}} = \frac{1}{N} \sum_i^{\mathrm{SV}} \Theta \left(1 - \frac{\hat{\alpha}_i}{\left[\mathbf{C}_{\mathrm{SV}}^{-1}\right]_{ii}} \right) \ , \tag{50}$$

where the sum goes over support vectors only. The non-support vectors will therefore not contribute to the leave-one-out error estimate.

An alternative derivation of eq. (50) is possible along the lines of section 4.1. Assuming that upon removal of an example the changes for all other $\hat{\alpha}_i$ will be so small that the remaining patterns will not change type, i.e. SVs will remain SVs, and non-SVs will remain non-SVs, the calculation of these changes reduce to a linear problem, resulting in eq. (50). These arguments may also be generalized to the non-separable case with linear slack penalty (Opper & Winther, 1999b). A similar type of loo estimator for SVM (for a different loss function) has been derived previously by Wahba (1998).

7 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) The iterative scheme used for solving the mean field equations are summarized in Appendix C. 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 D. 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 D) and for the mean field algorithms the additional output noise parameter κ (described in Appendix A). 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 D) 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 ϵ_{loo}. 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 four 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 8, we briefly describe how the Bayesian 'evidence' (also called ML-II) model selection framework (Mackay, 1992) may be used within the mean field framework. In a forthcoming paper, we will discuss these issues in detail and discuss how to automate the hyperparameter determination to make it possible to work with a much larger number of free (hyper)parameters.

Test of leave-one-out estimators. The generalization error estimators 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 approximations of the leave-one-out estimators are very precise. Usually they get at most one classification wrong compared to the exact result. The single case where a complete failure was observed, was for the extremely short ranged covariance function in figure 2.

7.1 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 roughly 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. The transmitted sonar signal is a frequency-modulated chirp, rising in amplitude. The data set contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock. Here we have used the same training/test data split as Gorman and Sejnowski, which may be obtained from
`http://www.boltz.cs.cmu.edu/benchmarks/sonar.html`.

Results. For all algorithms the number of hyperparameters were strongly reduced by setting $\mathbf{w} = 1/\sigma^2 N$. 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.

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
Back-Prop SP	0.269		
Best 2layer 12H	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.

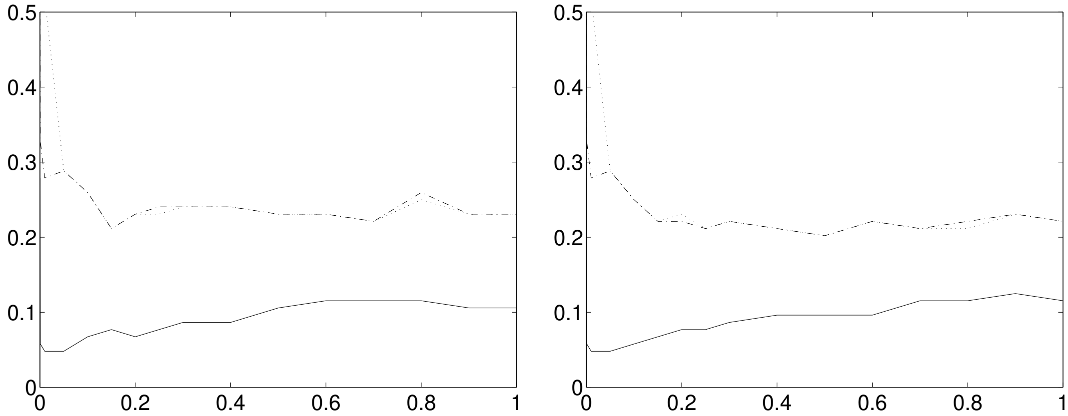

Figure 2: The correlation between the test error ϵ_{test} (full line) and the leave-one-out estimator ϵ_{loo} (dotted line) and $\epsilon_{\text{loo}}^{\text{exact}}$ (dash-dotted line) as a function of σ^2. The left plot is for TAP and the right plot is for SVM. Note that the test error increases steeply for very small values of σ.

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 (Pontil & Verri, 1998). The results show a clear correlation between the solutions found by the three algorithms. However, the mean field approaches have a tendency to have a much more skewed distribution of stabilities.

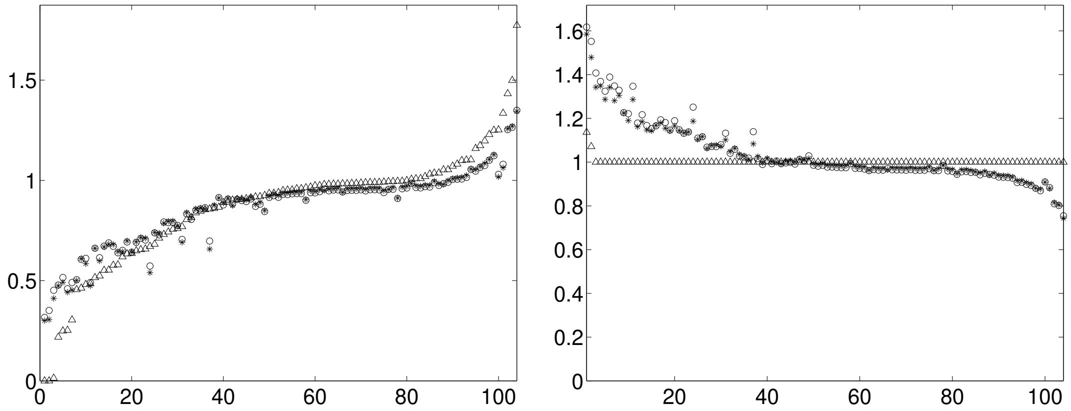

Figure 3: Left figure: The 'embedding strengths' α_i plotted for the different algorithms for the Sonar data. The right figure: The mean activation $t_i\langle h_i \rangle$ for the different algorithms (same ordering as the left plot). The triangles are for support vectors, circles are for naive mean 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.

7.2 Leptograpsus Crabs and Pima Indians Diabetes

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

For 'Leptograpsus Crabs' the task is to classify the sex of crabs on the basis of five anatomical attributes and a color attribute. The 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 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), 'Laplace GP' and all other are from Ripley (1994,1996). Williams

& Barber (1997) studied both ML-II and hybrid Monte Carlo (HMC) integration to deal with the hyperparameters. Our results found are close to the state of the art. The performance is quite sensitive to the choice of **w**-hyperparameters so, hopefully, one may 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 we tried give rise to similar performance.

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

	Leptograpsus Crabs				Pima Indians Diabetes			
Algorithm	Err.	ϵ_{test}	$\epsilon_{\text{loo}}^{\text{exact}}$	ϵ_{loo}	Err.	ϵ_{test}	$\epsilon_{\text{loo}}^{\text{exact}}$	ϵ_{loo}
TAP Mean Field	4	0.033	0.037	0.037	63	0.190	0.250	0.255
Naive Mean Field	2	0.017	0.025	0.025	64	0.193	0.245	0.245
SVM	6	0.050	0.037	0.037	66	0.199	0.250	0.250
Laplace GP–HMC	3				68			
–ML-II	4				69			
Best 2layer Network	3				75			
Simple Perceptron	8				67			
Logistic Regression	4				66			
MARS (degree=1)	8				75			
PP regression (4rf)	3				75			

8 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 modelling. 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.

We have further shown how support vector machines (SVM) can be obtained from a maximum a posteriori limit of the Gaussian process model. The built-in leave-one-out estimators for the generalization error provided by the mean field method naturally extend to the SVM 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, 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). 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 costfunction 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 (for certain smaller datasets) 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 computationally simpler evidence framework for hyperparameter selection is used, the naive mean field theory might give 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.

Acknowledgments

We are thankful to Pál Ruján, Sara A. Solla and Grace Wahba for discussions. This research is supported by the Swedish Foundation for Strategic Research.

A Noise Models

In this appendix, we discuss an input noise and an output noise model and the modification of the mean field algorithms implied.

Gaussian input noise. Input noise is here defined as a random term added to the activation function in the likelihood:

$$p(t|h(\mathbf{x}), \xi(\mathbf{x})) = \Theta(t(h(\mathbf{x}) + \xi(\mathbf{x}))) \tag{51}$$

We will assume that the noise is Gaussian with zero mean and variance v_2. With Gaussian noise, the original step function likelihood will be modified to a sigmoidal shaped 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. Both formulations give equivalent mean field equations for both TAP and 'naive' mean field theory. However, the interpretation of some the quantities in the theory change, 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. Both α_i and importantly for the leave-one-out estimate $\langle h_i \rangle_{\setminus i}$ are the same in both formulations.

Output noise. The output noise is flip noise, i.e.

$$p(t|h(\mathbf{x}), \xi(\mathbf{x})) = \Theta(t\ \xi(\mathbf{x})\ h(\mathbf{x})) \tag{52}$$

where $\xi = \pm 1$. We will consider simple iid output flip noise with the flip probability given by κ. If we denote the Likelihood without output noise by $p(t|h)$, the Likelihood for output noise is

$$\begin{aligned} p^{\text{o.n.}}(t|h) &= \kappa p(-t|h) + (1-\kappa)p(t|h) \\ &= \kappa + (1-2\kappa)p(t|h)\ . \end{aligned} \tag{53}$$

The predictive probability eq. (7), $p^{\text{o.n.}}(t|\mathbf{t}) = \int dh p^{\text{o.n.}}(t|h)p(h|\mathbf{t})$ becomes

$$p^{\text{o.n.}}(t|\mathbf{t}) = \kappa + (1-2\kappa)p(t|\mathbf{t})\ , \tag{54}$$

The embedding function α_i eq. (16) is modified accordingly

$$\alpha_i = \frac{1}{\sqrt{\lambda_i}} \frac{(1-2\kappa)D\left(\frac{\langle h_i \rangle_{\setminus i}}{\sqrt{\lambda_i}}\right)}{\kappa + (1-2\kappa)\Phi\left(t_i \frac{\langle h_i \rangle_{\setminus i}}{\sqrt{\lambda_i}}\right)}\ . \tag{55}$$

The expression for Λ_i eq. (28) which depends on a derivative of α_i turns out to unchanged compared to the noiseless case eq. (59).

B Summary Mean Field Equations

In this appendix, the TAP mean field algorithm, the Bayes classifier and the leave-one-out estimator for the generalization error are given. All the results are for noise-free classification. Generalization to different types of noise may be found in Appendix A.

$$\alpha_i = \frac{1}{\sqrt{\lambda_i}} \frac{D(z_i)}{\Phi(z_i)} \qquad z_i = t_i \frac{\langle h_i \rangle_{\setminus i}}{\sqrt{\lambda_i}} \tag{56}$$

$$\langle h_i \rangle_{\setminus i} = \langle h_i \rangle - \lambda_i t_i \alpha_i \tag{57}$$

$$\langle h_i \rangle = \sum_j C_{ij} t_j \alpha_j \tag{58}$$

$$\Lambda_i = \lambda_i \left(\frac{1}{t_i \alpha_i \langle h_i \rangle} - 1 \right) \tag{59}$$

$$\lambda_i = \frac{1}{\left[(\mathbf{\Lambda} + \mathbf{C})^{-1} \right]_{ii}} - \Lambda_i \tag{60}$$

The Bayes classifier:

$$t^{\mathrm{Bayes}}(D_m, \mathbf{x}) = \mathrm{sgn} \langle h(\mathbf{x}) \rangle \tag{61}$$

with $\langle h(\mathbf{x}) \rangle = \sum_i C(\mathbf{x}, \mathbf{x}_i) t_i \alpha_i$.

Leave-one-out estimators for the generalization error:

$$\epsilon_{\mathrm{loo}} = \frac{1}{N} \sum_i \Theta(-t_i \langle h_i \rangle_{\setminus i}) \tag{62}$$

C Solving the Mean Field Equations

The mean field equations are solved by iteration. The iterative scheme used are described below in pseudo-code. The iterative scheme for the naive mean field is obtained by omitting the statements for Λ_i and λ_i in the iteration step.

The most expensive step of 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 find the α_i-variables than to find λ_i and Λ_i. We therefore choose to update Λ_i and 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 for takes about 1 CPU second on an Alpha 433au for the largest problem studied ($N = 200$). An important contributing factor to learning speed is to an iterative learning rate: We set $\eta \to 1.1\eta$ if 'the error' $\sum_i |\delta\alpha_i|^2$ decreases in the update step and $\eta \to \eta/2$ if it increases.

Initialization.

1. $\eta = 0.05$.

2. $\mathrm{ftol} = 0.00001$.

3. Calculate covariance matrix $\mathbf{C}$.

4. For every training example i do:

(a) $\alpha_i = 0.0$.

(b) $\lambda_i = C_{ii}$.

Iterate. While $\max_i(|\delta\alpha_i|^2) > \text{ftol}$ do:

1. For every training example i do: $\alpha_i = \alpha_i + \eta\delta\alpha_i$

2. For every 20th iteration do for every i:

   (a) $\Lambda_i = \lambda_i \left(\frac{1}{\langle h_i \rangle t_i \alpha_i} - 1\right)$.

   (b) $\lambda_i = \frac{1}{[(\mathbf{\Lambda}+\mathbf{C})^{-1}]_{ii}} - \Lambda_i$.

Definitions

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

D Covariance Functions

Covariance functions. A thorough discussion of covariance functions and their properties is given in Abrahamsen (1997). We have tested the four covariance functions given below. The first two is obtained from feed-forward neural network models which converge to Gaussian processes. In all three cases the priors on the weights and thresholds are independent Gaussians. The first corresponds to the simple perceptron and the second is found from taking the limit of infinite number of hidden units with sigmoidal (error-function) activation functions (Neal, 1996; Williams, 1997). The v_1-term is the output-layer threshold variance. 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).

1. **Simple Perceptron**

$$C(\mathbf{x}, \mathbf{x}') = \sum_{l=1}^{d} w_l x_l x_l' + v_1 \tag{63}$$

2. **Infinite Network Sigmoid-Activation Function**

$$C(\mathbf{x}, \mathbf{x}') = v_0 \frac{2}{\pi} \arcsin \left(\frac{\sum_l w_l x_l x_l'}{\sqrt{(1 + \sum_l w_l x_l x_l)(1 + \sum_l w_l x_l' x_l')}} \right) + v_1 \tag{64}$$

To get a covariance function corresponding to an infinite network with sign activation function, the factors of one should be removed from the expression.

3. **Gaussian**

$$C(\mathbf{x}, \mathbf{x}') = v_0 \exp\left(-\frac{1}{2}\sum_l w_l (x_l - x'_l)^2\right) + v_1 \tag{65}$$

4. **Cauchy**

$$C(\mathbf{x}, \mathbf{x}') = \frac{v_0}{\left(1 + \sum_l w_l (x_l - x'_l)^2\right)^\delta} + v_1 \tag{66}$$

with $\delta = 1$ in all simulations.

The length scale for input dimension l is defined by $1/w_l$. Important inputs will be well-defined and thus be associated with a relative short length scale and vice versa for unimportant length scales. All these choices correspond to an *a priori* expectation that the inputs are independent, because there are no correlation terms of the type $w_{ij} s_i s'_j$. It is possible to estimate correlations in the data (Vivarelli & Williams, 1999). Since the properties of the solution is independent of the scale of Gaussian fields we may fix v_0 to say $v_0 = 1$.

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).

J. H. Friedman and J. W. Tukey, A Projection Pursuit Algorithm for Exploratory Data Analysis, IEEE Transactions on Computers **23**, 881-890 (1974).

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

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 CRG-TR-97-2, Dept. of Computer Science, University of Toronto (1997).

M. Opper, E. Fokue, and O. Winther, in preparation (1999).

M. Opper and O. Winther, A Mean Field Approach to Bayes Learning in Feed-Forward Neural Networks, Rhys. 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 Process Classification and SVM: Mean Field Results, in *Large Margin Classifiers*, P. Bartlett, B. Schoelkopf, 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).

M. Pontil and A. Verri, Properties of Support Vector Machines, Neural Comp. **10** 953-974 (1998).

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. Schoelkopf, 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).

F. Vivarelli and C. K. I. Williams, Discovering Hidden Features with Gaussian Process Regression, 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, (1999).

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 (Schoelkopf, 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).