Journal of Machine Learning Research 4 (2003) 1151-1173 Submitted 5/03; Published 12/03

An Approximate Analytical Approach to Resampling Averages

Dörthe Malzahn MALZAHND@ISP.IMM.DTU.DK
Informatics and Mathematical Modelling
Technical University of Denmark
R.-Petersens-Plads, Building 321
Lyngby, DK-2800, Denmark
and
Institute of Mathematical Stochastics
University of Karlsruhe
Englerstr. 2
Karlsruhe, D-76131, Germany

Manfred Opper OPPERM@ASTON.AC.UK
School of Engineering and Applied Science
Aston University
Birmingham, B4 7ET, United Kingdom

Editor: Christopher K. I. Williams

Abstract

Using a novel reformulation, we develop a framework to compute approximate resampling data averages analytically. The method avoids multiple retraining of statistical models on the samples. Our approach uses a combination of the replica "trick" of statistical physics and the TAP approach for approximate Bayesian inference. We demonstrate our approach on regression with Gaussian processes. A comparison with averages obtained by Monte-Carlo sampling shows that our method achieves good accuracy.

Keywords: bootstrap, kernel machines, Gaussian processes, approximate inference, statistical physics

1. Introduction

Resampling is a widely applicable technique in statistical modeling and machine learning. By resampling data points from a single given set of data one can create many new data sets which allows to simulate the effects of statistical fluctuations of parameter estimates, predictions or any other interesting function of the data. Resampling is the basis of Efron's *bootstrap* method which is a general approach for assessing the quality of statistical estimators (Efron, 1979, Efron and Tibshirani, 1993). It is also an essential part of the *bagging* and *boosting* approaches in machine learning where the method is used to obtain a better model by averaging different models which were trained on the resampled data sets.

In this paper, we will not provide theoretical foundations of resampling methods, nor do we intend to give a critical discussion of their applicability. Interested readers are referred to standard literature (such as Efron, 1982, Efron and Tibshirani, 1993, Shao and Tu, 1995). The main goal of the paper is to present a novel method for dealing with the large *computational complexity* that can present a significant technical problem when resampling methods are applied to complex statistical machine-learning models on large data sets.

To explain the resampling method in a fairly general setting, we assume a given sample $D_0 = (z_1, z_2, \ldots, z_N)$ of data points. For example, z_i might denote a pair (x_i, y_i) of inputs and output labels used to train a classifier. New artificial data samples D of arbitrary size S can be created by resampling points from D_0. Writing $D = (z'_1, z'_2, \ldots, z'_S)$ one chooses each z'_i to be an arbitrary point of D_0. Hence, some z_i in D_0 will appear multiple times in D and others not at all. A typical task in the resampling approach is the computation of certain *resampling averages*. Let $\theta(D)$ denote a quantity of interest which depends on the data sets D. We define its resampling average by

$$E_{D\sim D_0}[\theta(D)] = \sum_{D\sim D_0} W(D)\,\theta(D) \tag{1}$$

where $\sum_{D\sim D_0}$ denotes a sum over all sets D generated from D_0 using a specific sampling method and $W(D)$ denotes a normalized weight assigned to each sample D. If the model is sufficiently complex (for example a *support-vector* machine, see e.g., Schölkopf et al., 1999), the retraining on each sample D to evaluate $\Theta(D)$ and averaging can be rather time consuming even when the total sum in Equation (1) is approximated by a random subsample using a Monte-Carlo approach. Hence, it is useful to develop *analytical* approximation techniques which avoid the repeated retraining of the model. Existing analytical approximations (based on asymptotic techniques) found, e.g., in the bootstrap literature such as the *delta* method and the *saddle-point* method usually require explicit analytical formulas for the quantities $\theta(D)$ that we wish to average (see e.g., Shao and Tu, 1995). These will usually not be available for more complex models in machine learning.

In this paper, we introduce a novel approach for the approximate calculation of resampling averages. It is based on a combination of three ideas. We first utilize the fact that often many interesting functions $\theta(D)$ can be expressed in terms of basic statistical estimators for parameters of certain statistical models. These can be *implicitly* defined as pseudo Bayesian expectations with suitably defined posterior Gibbs distributions over model parameters. Hence, the method does not require an *explicit* analytical expression for these statistics. Within our formulation, it becomes possible to exchange posterior expectations and data averages and perform the latter ones *analytically* using the so-called "replica trick" of statistical physics (Mézard et al., 1987). After the data average, we are left with a typically intractable inference problem for an effective Bayesian probabilistic model. As a final step, we use techniques for approximate inference to treat the probabilistic model. This combination of techniques allows us to obtain approximate resampling averages by solving a set of nonlinear equations rather than by explicit sampling. We demonstrate the method on bootstrap estimators for *regression with Gaussian processes (GP)* (which is a kernel method that has gained high popularity in the machine-learning community in recent years, see Neal, 1996) and compare our analytical results with results obtained by Monte-Carlo sampling.

The paper is organized as follows: Section 2 presents the key ideas of our theory in a general setting. Section 3 discusses bootstrap in the context of our theory, that is we specialize to the case that the data sets D are obtained from D_0 by independent sampling with replacement. In Section 4, we derive general formulas for interesting resampling averages of GP models such as the generalization error and the mean and variance of the prediction. In Section 5, we apply the results of Section 3 and 4 to the bootstrap of a GP regression model. Section 6 concludes the paper with a summary and a discussion of the results.

2. Outline of the Basic Ideas

This section presents the three key ideas of our approach for the approximate analytical calculation of resampling averages.

2.1 Step I: Deriving Estimators from Gibbs Distributions

Our formalism assumes that the functions $\theta(D)$ which we wish to average over data sets can be expressed in terms of a set of basic statistics $\hat{\mathbf{f}}(D) = (\hat{f}_1(D), \ldots, \hat{f}_M(D))$ of the data. $\hat{\mathbf{f}}(D)$ can be understood as an *estimator* for a parameter vector $\mathbf{f}$ which is used in a statistical model describing the data. To be specific, we assume that $\theta(D)$ can be expanded in a formal multivariate power series expansion which we write symbolically as

$$\theta(D) = \sum_r \mathbf{c}_r \hat{\mathbf{f}}(D)^r , \tag{2}$$

where $\hat{\mathbf{f}}(D)^r$ stands for a collection of terms of the form $\prod_{k=1}^r \hat{f}_{i_k}$ and the i_k's are indices from the set $\{1, \ldots, M\}$. $\mathbf{c}_r$ denotes a collection of corresponding expansion coefficients.

Our crucial assumption is that the basic estimators $\hat{\mathbf{f}}(D)$ can be written as posterior expectations

$$\hat{\mathbf{f}}(D) = \langle \mathbf{f} \rangle = \int d\mathbf{f}\, \mathbf{f}\, P(\mathbf{f}|D) \tag{3}$$

with a posterior density

$$P(\mathbf{f}|D) = \frac{1}{Z(D)} \mu(\mathbf{f})\, P(D|\mathbf{f}) \tag{4}$$

that is constructed from a suitable prior distribution $\mu(\mathbf{f})$ and a likelihood term $P(D|\mathbf{f})$.

$$Z(D) = \int d\mathbf{f}\, \mu(\mathbf{f})\, P(D|\mathbf{f}) \tag{5}$$

denotes a normalizing partition function. We will denote expectations with respect to Equation (4) by angular brackets $\langle \cdots \rangle$. This representation avoids the problem of writing down explicit, complicated formulas for $\hat{\mathbf{f}}$. Our choice of Equation (3) obviously includes Bayesian (point) estimators of model parameters, but with specific choices of likelihoods and priors maximum-likelihood and maximum-a-posteriori estimators can also be covered by the formalism.

From the expansion Equation (2) and the linearity of the data averages, it seems reasonable to reduce the computation of the average $E_{D\sim D_0}[\theta(D)]$ to that of averaging the simple monomials $\hat{\mathbf{f}}(D)^r$ and try a resummation of the averaged series at the end. Using Equation (3) we can write

$$E^{(r)} \doteq E_{D\sim D_0}[\hat{\mathbf{f}}^r] = E_{D\sim D_0}[\langle \mathbf{f} \rangle^r] = E_{D\sim D_0}\left[\frac{1}{Z(D)^r} \int \prod_{a=1}^r \{d\mathbf{f}^a\, \mathbf{f}^a\, \mu(\mathbf{f}^a)\, P(D|\mathbf{f}^a)\}\right] . \tag{6}$$

which involves r copies, i.e., *replicas* $\mathbf{f}^a$ for $a = 1, \ldots, r$ of the parameter vector $\mathbf{f}$. The superscripts should NOT be confused with powers of the variables.

2.2 Step II: Analytical Resampling Average Using the Replica Trick

To understand the simplifications which can be gained by our representation Equation (3), one should note that in a variety of interesting and practically relevant cases it is possible to compute resampling averages of the type $E_{D\sim D_0}[\prod_{a=1}^r P(D|\mathbf{f}^a)]$ analytically in a reasonably simple form. Hence, if the partition functions $Z(D)$ in the denominator of Equation (6) were absent, or would not depend on D, one could easily exchange the Bayes average with the data average and would be able to get rid of resampling averages in an analytical way. One would then be left with a *single* Bayesian type of average which could be computed by other tools known in the field of probabilistic inference.

To deal with the unpleasant partition functions $Z(D)$ to enable an analytical average over data sets (which is the "quenched disorder" in the language of statistical physics) one introduces the following "trick" extensively used in statistical physics of amorphous systems (Mézard et al., 1987). We introduce the auxiliary quantity

$$E_n^{(r)} \doteq E_{D\sim D_0}\left[Z(D)^{n-r}\int\prod_{a=1}^{r}\{d\mathbf{f}^a\,\mu(\mathbf{f}^a)\,P(D|\mathbf{f}^a)\mathbf{f}^a\}\right]$$

for arbitrary real n, which allows to write

$$E^{(r)} = \lim_{n\to 0} E_n^{(r)}.$$

The advantage of this definition is that for *integers* $n \geq r$, the partition functions $Z(D)$ in $E_n^{(r)}$ can be eliminated by using a total number of n *replicas* $\mathbf{f}^1, \mathbf{f}^2, \ldots, \mathbf{f}^n$ of the original variable $\mathbf{f}$. Using the explicit form of the partition function $Z(D)$, Equation (5), we get

$$E_n^{(r)} = E_{D\sim D_0}\left[\int\prod_{a=1}^{n}\{d\mathbf{f}^a\,\mu(\mathbf{f}^a)\,P(D|\mathbf{f}^a)\}\prod_{a=1}^{r}\mathbf{f}^a\right]$$

Now, we can exchange the expectation over data sets with the expectation over $\mathbf{f}$'s and obtain

$$E_n^{(r)} = \Xi_n\left\langle\left\langle\prod_{a=1}^{r}\mathbf{f}^a\right\rangle\right\rangle \tag{7}$$

where $\langle\langle\cdots\rangle\rangle$ denotes an average with respect to a new Gibbs measure $P(\mathbf{f}^1,\ldots,\mathbf{f}^n|D_0)$ for replicated variables which results from the data average. It is defined by

$$P(\mathbf{f}^1,\ldots,\mathbf{f}^n|D_0) = \frac{1}{\Xi_n}\left(\prod_{a=1}^{n}\mu[\mathbf{f}^a]\right)P(D_0|\mathbf{f}^1,\ldots,\mathbf{f}^n) \tag{8}$$

with likelihood

$$P(D_0|\mathbf{f}^1,\ldots,\mathbf{f}^n) = E_{D\sim D_0}\left[\prod_{a=1}^{n}P(D|\mathbf{f}^a)\right] \tag{9}$$

and normalizing partition function Ξ_n. Since by construction $\lim_{n\to 0}\Xi_n = 1$, we will omit factors Ξ_n in the following.

2.3 Step III: Approximate Inference for the Replica Model

We have mapped the original problem of computing a resampling average to an inference problem with a Bayesian model, where the hidden variables have the dimensionality $M \times n$ and n must be set to zero at the end. Of course, we should not expect to be able to compute averages over the measure Equation (8) analytically, otherwise we would have found an exact solution to the resampling problem. Our final idea is to resort to techniques for *approximate inference* (see e.g., Opper and Saad, 2001) which have recently become popular in machine learning. Powerful methods are the *variational Gaussian approximation*, the *mean field method*, the *Bethe approximation* and the *adaptive TAP* approach. They have in common that they approximate intractable averages by integrations over tractable distributions which contain specific optimized parameters. We found that for these methods, the "replica limit" $n \to 0$ can be performed *analytically before* the final numerical parameter optimization. Note that the measure Equation (8) (which we will approximate) characterizes the *average properties* of the learning algorithm with respect to the ensemble of training data sets $D \sim D_0$. We do NOT approximate the individual predictors $\hat{f}(D)$.

3. Independent Sampling with Replacement

Often, statistical models of interest assume likelihoods which are factorizing in the individual data points, that is

$$P(D_0|\mathbf{f}) = \prod_{j=1}^{N} \exp\left(-h(\mathbf{f}, z_j)\right)$$

where h is a type of "training error". Each new sample $D \sim D_0$ can be represented by a vector of "occupation" numbers $\mathbf{s} = (s_1, \ldots, s_N)$ where s_i is the number of times example z_i appears in the set D and we require $\sum_{i=1}^{N} s_i = S$, where S is the fixed size of the data sets. In this case we can write

$$P(D|\mathbf{f}) = \prod_{j=1}^{N} \exp\left(-s_j h(\mathbf{f}, z_j)\right) \tag{10}$$

and the resampling average $E_{D \sim D_0}$ becomes simply an average over the distribution of occupation numbers.

In the remainder of this paper, we specialize to the important case of an *independent resampling* of each data point *with replacement* used in the bootstrap (Efron, 1979) and bagging (Breiman, 1996) approaches. Each data point z_j in D_0 is chosen with equal probability $1/N$ to become an element of D. The statistical weight $W(D) \to W(\mathbf{s})$ for a sample D represented by the vector $\mathbf{s}$ in the resampling averages Equation (1) can be obtained from the fact that the distribution of s_i's is multinomial. However, it is simpler (and does not make a big difference when the sample size S is sufficiently large) when we also randomize the sample sizes by using a Poisson distribution for S. In the following, the variable S will denote the *mean number of data points* in the samples. In this case we get the simpler, factorizing weight for the samples given by

$$W(\mathbf{s}) = \prod_{j=1}^{N} \frac{\left(\frac{S}{N}\right)^{s_j} e^{-S/N}}{s_j!} \tag{11}$$

Using the explicit form of the distribution Equation (11) and of the likelihood Equation (10), the new likelihoods Equation (9)

$$P(D_0|\mathbf{f}^1,\ldots,\mathbf{f}^n) = E_{D\sim D_0}\left[\prod_{a=1}^{n} P(D|\mathbf{f}^a)\right] = \sum_{\mathbf{s}} W(\mathbf{s})\left[\prod_{j=1}^{N} e^{-s_j \sum_{a=1}^{n} h(\mathbf{f}^a,z_j)}\right]$$

will again factorize in the data points and we get

$$P(D_0|\mathbf{f}^1,\ldots,\mathbf{f}^n) = \prod_{j=1}^{N} L_j(\mathbf{f}^1,\ldots,\mathbf{f}^n) \tag{12}$$

with the local likelihood

$$L_j(\mathbf{f}^1,\ldots,\mathbf{f}^n) = \exp\left(-\frac{S}{N}\left(1-\prod_{a=1}^{n} e^{-h(\mathbf{f}^a,z_j)}\right)\right) . \tag{13}$$

We will continue the discussion for the example of the bootstrap. Note however, that the following results can be applied to other sampling schemes as well by using a suitable factorizing distribution $W(\mathbf{s}) = \prod_{j=1}^{N} p(s_j)$ and replacing Equation (13) by the respective expression for the likelihood.

4. Resampling Averages for Gaussian Process Models

We will apply our approach to the computation of bootstrap estimates for a variety of quantities related to Gaussian process predictions.

4.1 Definition of Gaussian Process Models

For Gaussian process (GP) models, the Bayesian framework of Section 2.1 is the natural choice where the vector $\mathbf{f}$ represents the values of an unknown function f at the input points of the data D_0, that is $\mathbf{f} = (f_1, f_2, \ldots, f_N)$ with $f_i \doteq f(x_i)$. The prior measure $\mu(\mathbf{f})$ is an N dimensional joint Gaussian distribution of the form

$$\mu(\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^N|\mathbf{K}|}} \exp\left[-\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\right] , \tag{14}$$

where the *kernel matrix* $\mathbf{K}$ has matrix elements $K(x_i,x_j)$, which are defined through the *covariance kernel* $K(x,x')$ of the process. For supervised learning problems, each data point $z_j = (x_j,y_j)$ consists of the input x_j (usually a finite dimensional vector) and a real label y_j. We will assume that the training error function $h(\mathbf{f},z_j)$ is local, i.e., it depends on the vector $\mathbf{f}$ only through the function value f_j. Hence, we will write

$$h(\mathbf{f},z_j) \to h(f_j,z_j) \tag{15}$$

in the following. The vector $\hat{\mathbf{f}}(D)$ represents the posterior mean prediction of the unknown function at the inputs x_i for $i = 1,\ldots,N$. For any choice $D \sim D_0$, some of these inputs will also appear in the training set D while others can be used as *test* inputs.

4.2 Resampling Averages of Local Quantities

Let us begin with simple resampling averages of the form $E_{D\sim D_0}\left[\hat{f}_i(D)\right]$ and $E_{D\sim D_0}\left[(\hat{f}_i(D))^2\right]$, from which one can estimate the bias and variance of the i-th component of the GP prediction. These averages can be directly translated into the replica formalism presented by Equation (6) and Section 2.2. We get

$$\begin{aligned} E_{D\sim D_0}\left[\hat{f}_i(D)\right] &= \lim_{n\to 0}\langle\langle f_i^1\rangle\rangle \\ E_{D\sim D_0}\left[(\hat{f}_i(D))^2\right] &= \lim_{n\to 0}\langle\langle f_i^1 f_i^2\rangle\rangle \end{aligned} \tag{16}$$

where the superscripts on the right hand side are replica indices and $\langle\langle\cdots\rangle\rangle$ denotes an average with respect to the Gibbs measure Equation (8) for replicated variables.

A somewhat more complicated example is *Efron's estimator for the bootstrap generalization error* of the predictor $\hat{\mathbf{f}}(D)$, Equation (3), where we specialize to the square error for testing

$$\varepsilon(S) \doteq \frac{1}{N}\sum_{i=1}^{N}\frac{E_{D\sim D_0}\left[\delta_{s_i,0}\left(\hat{f}_i(D)-y_i\right)^2\right]}{E_{D\sim D_0}\left[\delta_{s_i,0}\right]} \,. \tag{17}$$

Equation (17) computes the average bootstrap test error at each data point i from D_0. The Kronecker symbol, defined by $\delta_{s_i,0}=1$ for $s_i=0$ and 0 else, guarantees that only realizations of training sets D contribute which do not contain the test point. Its occurrence requires a small change in our basic formalism. A simple calculation shows that the effect of the term $\delta_{s_i,0}$ in the resampling average is the replacement of the i-th local likelihood L_i, Equation (13), in the product Equation (12) by one. Hence, with a slight generalization of Equation (7) we have

$$\frac{E_{D\sim D_0}\left[\delta_{s_i,0}\,(\hat{f}_i(D)-y_i)^2\right]}{E_{D\sim D_0}\left[\delta_{s_i,0}\right]} = \lim_{n\to 0}\left\langle\!\!\left\langle\frac{(f_i^1-y_i)(f_i^2-y_i)}{L_i(f_i^1,\ldots,f_i^n)}\right\rangle\!\!\right\rangle \tag{18}$$

Note that with Equation (15) the local likelihoods of Equation (13) can be simplified, replacing $L_i(\mathbf{f}^1,\ldots,\mathbf{f}^n)$ with $L_i(f_i^1,\ldots,f_i^n)$.

4.3 Approximate Inference for the Replica Model Using the ADATAP Approach

To deal with the intractable Bayesian averages in Equations (16) and (18) we have used the *variational Gaussian approximation* (VG), the *mean field approximation* (MF) and the *adaptive TAP* approach (ADATAP). Since the graph of the probabilistic model corresponding to GP's is fully connected we did refrain from using the *Bethe approximation*. We found that the ADATAP approach of Opper and Winther (2000, 2001a,b) was the most suitable technique which gave superior performance compared to the VG and MF approximations. Hence, we will give the explicit analytical derivations only for the ADATAP method, but will present some numerical results for the performance of the other techniques.

An important simplification in the computation of Equations (16) and (18) comes from the fact that these are local averages which depend only on the replicated variables f_i^a for a single data point i and can be computed from the knowledge of the *marginal distribution* $P_i(\vec{f}_i)$ alone, where we have introduced the n-dimensional vectors

$$\vec{f}_i = (f_i^1,\ldots,f_i^n)$$

for $i = 1,\dots,N$. The ADATAP approximation presents a selfconsistent approximation[1] of marginal distributions $P_i(\vec{f}_i)$ for $i = 1,\dots,N$. It is based on factorizing

$$P_i(\vec{f}) = \frac{L_i(\vec{f})\, P_{\setminus i}(\vec{f})}{\int d\vec{f}\, L_i(\vec{f})\, P_{\setminus i}(\vec{f})} \tag{19}$$

where the *cavity distribution* is defined as

$$P_{\setminus i}(\vec{f}_i) \propto \int \prod_{j=1, j\neq i}^{N} d\vec{f}_j \prod_{a=1}^{n} \mu(\mathbf{f}^a) \prod_{j=1, j\neq i}^{N} L_j(\vec{f}_j)\ . \tag{20}$$

It represents the influence of all variables $\vec{f}_j = (f_j^1,\dots,f_j^n)$ with $j \neq i$ on the variable $\vec{f}_i$. Following Opper and Winther (2000) (slightly generalizing the original idea to vectors of n variables), the cavity distribution Equation (20) is approximated by a Gaussian distribution, i.e., a density of the form

$$P_{\setminus i}(\vec{f}) = \frac{1}{Z_n(i)}\, e^{-\frac{1}{2}\vec{f}^T \Lambda_c(i)\vec{f} + \vec{\gamma}_c(i)^T \vec{f}}\ . \tag{21}$$

To compute the parameters $\Lambda_c(i)$ and $\vec{\gamma}_c(i)$ in the N approximated cavity distributions Equation (21) self-consistently, one assumes that these are, independently of the local likelihood functions, entirely determined by the values of the first two *marginal moments*

$$\begin{aligned} \langle\langle \vec{f}_i \rangle\rangle &= \frac{\int d\vec{f}\, \vec{f}\, L_i(\vec{f})\, P_{\setminus i}(\vec{f})}{\int d\vec{f}\, L_i(\vec{f})\, P_{\setminus i}(\vec{f})} \\ \langle\langle \vec{f}_i\, \vec{f}_i^T \rangle\rangle &= \frac{\int d\vec{f}\, \vec{f}\, \vec{f}^T\, L_i(\vec{f})\, P_{\setminus i}(\vec{f})}{\int d\vec{f}\, L_i(\vec{f})\, P_{\setminus i}(\vec{f})} \end{aligned} \tag{22}$$

for $i = 1,\dots,N$ where we used Equation (19). The set of parameters $\Lambda_c(i)$, $\vec{\gamma}_c(i)$ which correspond to the actual likelihood can then be computed using an alternative set of tractable likelihoods $\hat{L}_j$. For GP models, we choose $\hat{L}_j$ to be Gaussian

$$\hat{L}_j(\vec{f}) = e^{-\frac{1}{2}\vec{f}^T \Lambda(j)\vec{f} + \vec{\gamma}(j)^T \vec{f}}\ . \tag{23}$$

The set of parameters $\Lambda(j)$ and $\vec{\gamma}(j)$ in Equation (23) is chosen in such a way that the corresponding joint Gaussian distribution (with GP prior Equation (14))

$$P_G(\tilde{\mathbf{f}}) \propto \prod_{a=1}^{n} \mu(\mathbf{f}^a) \prod_{j=1}^{N} \hat{L}_j(\vec{f}_j) \tag{24}$$

has first two marginal moments

$$\begin{aligned} \langle\langle \vec{f}_i \rangle\rangle &= \int d\tilde{\mathbf{f}}\, \vec{f}_i\, P_G(\tilde{\mathbf{f}}) \\ \langle\langle \vec{f}_i\, \vec{f}_i^T \rangle\rangle &= \int d\tilde{\mathbf{f}}\, \vec{f}_i\, \vec{f}_i^T\, P_G(\tilde{\mathbf{f}}) \end{aligned} \tag{25}$$

1. For motivations and alternative derivations of the approximation, see Opper and Winther (2000, 2001a,b) and Csató et al. (2002).

that coincide with those computed in Equation (22) for the intractable distribution $P(\tilde{\mathbf{f}}|D_0)$, Equation (8), for $i = 1, \ldots N$. Here we have defined $\tilde{\mathbf{f}} \doteq (\mathbf{f}^1, \ldots, \mathbf{f}^n)$. Hence, using Equation (19) and the assumed independence of $P_{\setminus i}(\vec{f})$ on the likelihood, we get

$$\begin{aligned}
\langle\langle \vec{f}_i \rangle\rangle &= \frac{\int d\vec{f}\, \vec{f}\, \hat{L}_i(\vec{f})\, P_{\setminus i}(\vec{f})}{\int d\vec{f}\, \hat{L}_i(\vec{f})\, P_{\setminus i}(\vec{f})} \\
\langle\langle \vec{f}_i\, \vec{f}_i^T \rangle\rangle &= \frac{\int d\vec{f}\, \vec{f}_i\, \vec{f}_i^T\, \hat{L}_i(\vec{f})\, P_{\setminus i}(\vec{f})}{\int d\vec{f}\, \hat{L}_i(\vec{f})\, P_{\setminus i}(\vec{f})} \,.
\end{aligned} \tag{26}$$

The three sets of Equations (22), (25) and (26) determine the sets of parameters $\Lambda(i)$, $\vec{\gamma}(i)$, $\Lambda_c(i)$ $\vec{\gamma}_c(i)$ together with the sets of moments for $i = 1, \ldots, N$ within the ADATAP approach. Note that the integrals of Equations (25) and (26) are Gaussian and can be performed trivially.

4.4 The Replica Limit $n \to 0$

The most crucial obstacle in computing the parameters of the cavity distribution Equation (21), that is the $n \times n$ matrix $\Lambda_c(i)$ and the n dimensional vector $\vec{\gamma}_c(i)$, is the limit $n \to 0$. To deal with it, one imposes symmetry constraints on $\Lambda_c(i)$ and $\vec{\gamma}_c(i)$ which make the *number* of distinct parameters *independent* of n. This will imply a similar symmetry for the marginal moments and for the parameters $\Lambda(i)$ and $\vec{\gamma}(i)$. To be specific, by the symmetry (exchangeability) of all n components $f_i^1, \ldots, f_i^n$ for each vector $\vec{f}_i$ in the distribution, we will assume the simplest choice known as *replica symmetry*, that is

$$\begin{aligned}
\Lambda_c^{ab}(i) = \lambda_c(i) \quad &\text{for}\, a \neq b, \quad \Lambda_c^{aa}(i) = \lambda_c^0(i) \text{ for all } a \\
\Lambda^{ab}(i) = \lambda(i) \quad &\text{for}\, a \neq b, \quad \Lambda^{aa}(i) = \lambda^0(i) \text{ for all } a
\end{aligned} \tag{27}$$

and also $\gamma^a(i) = \gamma(i)$ and $\gamma_c^a(i) = \gamma_c(i)$ for all $a = 1, \ldots, n$. More complicated parameterizations are possible and even necessary in complex situations when multivariate distributions have a large number of modes with almost equal statistical weight (see Mézard et al., 1987).

Using the symmetry properties Equation (27), we can decouple the replica variables in Equations (25) and (26) by the following transformation

$$e^{-\frac{1}{2}\vec{f}^T \Lambda_c(i)\vec{f} + \vec{\gamma}_c(i)^T \vec{f}} = \int dG(u) \prod_{a=1}^{n} \left\{ e^{-\frac{\Delta\lambda_c(i)}{2}(f^a)^2 + f^a\left(\gamma_c(i) + u\sqrt{-\lambda_c(i)}\right)} \right\} \,. \tag{28}$$

We have defined $\Delta\lambda_c(i) = \lambda_c^0(i) - \lambda_c(i)$ and $dG(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}\, du$ as the standard normal Gaussian distribution.

4.5 Results for the ADATAP Parameters

The TAP approach computes simultaneously approximate values for the first two sets of marginal moments of the posterior Equation (8) of the replica model. With Equation (27) they obey the following symmetry constraints

$$\begin{aligned}
\langle\langle f_i^a \rangle\rangle = m_i \quad \text{and} \qquad & \langle\langle (f_i^a)^2 \rangle\rangle = M_i \;\; \text{for}\, a = 1, \ldots, n \\
& \langle\langle f_i^a f_i^b \rangle\rangle = Q_{ii} \;\; \text{for } a \neq b \text{ and } a, b = 1, \ldots, n.
\end{aligned} \tag{29}$$

We can interpret their values in the limit $n = 0$ (within the TAP approximation) in terms of averages over the bootstrap ensemble. With Equation(16) and Section 2.2

$$\begin{aligned} m_i &= E_{D\sim D_0}\left[\hat{f}_i(D)\right] \\ M_i &= E_{D\sim D_0}\left[\langle (f_i(D))^2\rangle\right] \\ Q_{ii} &= E_{D\sim D_0}\left[(\hat{f}_i(D))^2\right] \end{aligned}$$

To compute the values of m_i, M_i and Q_{ii}, we use the symmetry properties Equation (27) and decouple the replica variables in Equations (25) and (26) by a transformation of the type Equation (28). We perform the limit $n \to 0$ and solve the remaining Gaussian integrals. The set of equations (26) yields

$$M_i - Q_{ii} = \frac{1}{\Delta\lambda(i) + \Delta\lambda_c(i)} \tag{30}$$

$$m_i = \frac{\gamma(i) + \gamma_c(i)}{\Delta\lambda(i) + \Delta\lambda_c(i)} \tag{31}$$

$$Q_{ii} - m_i^2 = -\frac{\lambda(i) + \lambda_c(i)}{(\Delta\lambda(i) + \Delta\lambda_c(i))^2} \tag{32}$$

where $\Delta\lambda(i) = \lambda^0(i) - \lambda(i)$ and $\Delta\lambda_c(i) = \lambda_c^0(i) - \lambda_c(i)$. The equations (25) yield for GP models

$$M_i - Q_{ii} = (\mathbf{G})_{ii} \tag{33}$$

$$m_i = (\mathbf{G}\,\gamma)_i \tag{34}$$

$$Q_{ii} - m_i^2 = -\left(\mathbf{G}\,\mathrm{diag}(\lambda)\,\mathbf{G}\right)_{ii} \tag{35}$$

with the $N \times N$ matrix

$$\mathbf{G} = (\mathbf{K}^{-1} + \mathrm{diag}(\Delta\lambda))^{-1}\ . \tag{36}$$

To perform the integral in Equation (22) it is useful to expand the likelihood first into a power series[2] introducing the abbreviation $\nu = S/N$

$$L_i(\vec{f}_i) = \exp\left(-\nu\left(1 - \prod_{a=1}^{n} e^{-h(f_i^a, z_i)}\right)\right) = \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \prod_{a=1}^{n} e^{-kh(f_i^a, z_i)}\ .$$

After decoupling the variables by Equation (28) we can take the replica limit $n = 0$. By introducing the measure

$$P_i(f|u,k) = \frac{e^{-kh(f,z_i) - \frac{\Delta\lambda_c(i)}{2} f^2 + f(\gamma_c(i) + u\sqrt{-\lambda_c(i)})}}{\int df\ e^{-kh(f,z_i) - \frac{\Delta\lambda_c(i)}{2} f^2 + f(\gamma_c(i) + u\sqrt{-\lambda_c(i)})}} \tag{37}$$

we arrive at the following compact results

$$m_i = \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \int dG(u) \int df\ f\ P_i(f|u,k) \tag{38}$$

$$M_i = \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \int dG(u) \int df\ f^2\ P_i(f|u,k) \tag{39}$$

$$Q_{ii} = \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \int dG(u) \left(\int df\ f\ P_i(f|u,k)\right)^2 \tag{40}$$

2. Note that this series represents just the average over all possible values of the occupation number $k \doteq s_i$ (see Equation (11)). We may easily use a different series corresponding to some other resampling scheme.

For a variety of "training energy" functions $h(f_i, z_i)$, the integrals can be performed analytically.

While the parameters m_i, M_i, Q_{ii} give local bootstrap averages at specific data points i, it is possible to extend the approximation to correlations between *different* data points. One simply uses the full covariance matrix of the auxiliary distribution $P_G(\tilde{\mathbf{f}})$, Equation (24), in order to approximate the covariance matrix of the true replica posterior $P(\tilde{\mathbf{f}})$. We get similarly to Equation (35)

$$Q_{ij} \doteq E_{D\sim D_0}\left[\hat{f}_i(D)\hat{f}_j(D)\right] = -\left(\mathbf{G}\,\mathrm{diag}(\lambda)\mathbf{G}\right)_{ij} + m_i m_j \ . \tag{41}$$

With Equation (33), we can interpret the matrix $\mathbf{G}$, Equation (36) as a theoretical estimate of the average bootstrapped posterior covariance where

$$G_{ij} = E_{D\sim D_0}\left[\langle f_i(D) f_j(D)\rangle\right] - Q_{ij} \ .$$

4.6 Results for the Resampling Estimate of the Generalization Error

Specializing to Efron's estimator of the generalization error Equation (17) and its replica expression Equation (18), we see that the latter is immediately expressed by an average $\langle\langle\cdots\rangle\rangle_{\backslash i}$ with respect to the cavity distribution Equation (20)

$$\frac{E_{D\sim D_0}\left[\delta_{s_i,0}\,(\hat{f}_i(D)-y_i)^2\right]}{E_{D\sim D_0}\left[\delta_{s_i,0}\right]} = \lim_{n\to 0}\left\langle\left\langle (f_i^a - y_i)(f_i^b - y_i)\right\rangle\right\rangle_{\backslash i} \tag{42}$$

with replica indices a, b where $a \neq b$. Note the similarity to the computation of a leave-one-out estimate. Here however, the leave-one-out estimate has to be computed for the *replicated and averaged* system. Inserting Equation (21) with Equation (28) into Equation (42) yields

$$\begin{aligned}&\left\langle\left\langle (f_i^a - y_i))(f_i^b - y_i)\right\rangle\right\rangle_{\backslash i}\\ &= \frac{\int dG(u)\left(\int df\,(f-y_i)\,e^{-\frac{\Delta\lambda_c(i)}{2}f^2 + f(\gamma_c(i)+u\sqrt{-\lambda_c(i)})}\right)^2 (Z_i(u))^{n-2}}{\int dG(u)(Z_i(u))^n}\end{aligned} \tag{43}$$

where we have defined

$$Z_i(u) = \int df\, e^{-\frac{\Delta\lambda_c(i)}{2}f^2 + f(\gamma_c(i)+u\sqrt{-\lambda_c(i)})} \ .$$

In Equation (43), the number n of replicas appears in a form which allows continuation to values $n < 2$ and to perform the limit $n \to 0$

$$\begin{aligned}&\lim_{n\to 0}\left\langle\left\langle (f_i^a - y_i))(f_i^b - y_i)\right\rangle\right\rangle_{\backslash i}\\ &= \int dG(u)\left(\frac{1}{Z_i(u)}\int df\,(f-y_i)\,e^{-\frac{\Delta\lambda_c(i)}{2}f^2 + f(\gamma_c(i)+u\sqrt{-\lambda_c(i)})}\right)^2\end{aligned} \tag{44}$$

Solving the remaining Gaussian integrals, our approximation for the bootstrapped mean square generalization error becomes

$$\varepsilon(S) = \frac{1}{N}\sum_{i=1}^{N}\frac{(\gamma_c(i) - y_i\Delta\lambda_c(i))^2 - \lambda_c(i)}{\Delta\lambda_c(i)^2} \ . \tag{45}$$

As discussed in Section 2.1, we can extend the whole argument to yield results for bootstrapping alternative generalization errors measured by other loss functions $g(\hat{f}_x(D);x,y)$. Expanding the loss function in a Taylor series in the variable $(\hat{f}_x(D)-y)$, we can apply the replica method to individual terms like $e^{S/N}E_{D\sim D_0}[\delta_{s_i,0}(\hat{f}_i(D)-y_i)^r]$. This simply replaces the power 2 in Equation (44) by the power r. We can thus resum the Taylor expansion and obtain

$$\begin{aligned}\varepsilon_g(S) &= \frac{1}{N}\sum_{i=1}^{N}\frac{E_{D\sim D_0}\left[\delta_{s_i,0}\, g\left(\hat{f}_i(D);x_i,y_i\right)\right]}{E_{D\sim D_0}\left[\delta_{s_i,0}\right]} \\ &= \frac{1}{N}\sum_{i=1}^{N}\int dG(u)\, g\left(\frac{\gamma_c(i)+u\sqrt{-\lambda_c(i)}}{\Delta\lambda_c(i)};x_i,y_i\right) \qquad (46)\end{aligned}$$

4.7 Further Bootstrap Averages

In the following, we derive results for the resampling statistics of the posterior mean predictor $\hat{f}_x$ of the unknown function f at *arbitrary* inputs x.

4.7.1 Mean and Variance of the Predictor

As is well known (see e.g., Csató and Opper, 2002), posterior mean predictors for GP models at arbitrary inputs x can be expressed in the form

$$\hat{f}_x \doteq \langle f_x\rangle = \sum_{i=1}^{N}\alpha_i K(x,x_i)$$

where the set of α_i's is independent of x and can be computed from the distribution of the finite dimensional vector $\mathbf{f}=(f_1,\ldots,f_N)$ alone. Hence, the bootstrap mean and the second moments can be expressed as

$$\begin{aligned}E_{D\sim D_0}\left[\hat{f}_x(D)\right] &= \sum_{i=1}^{N}E_{D\sim D_0}[\alpha_i]K(x,x_i) \qquad (47)\\ E_{D\sim D_0}\left[\hat{f}_x(D)\hat{f}_{x'}(D)\right] &= \sum_{i,j=1}^{N}K(x,x_i)E_{D\sim D_0}[\alpha_i\alpha_j]K(x_j,x')\end{aligned}$$

Setting $m_l = E_{D\sim D_0}\left[\hat{f}_l(D)\right]$ and $Q_{lk}=E_{D\sim D_0}\left[\hat{f}_l(D)\hat{f}_k(D)\right]$ for arbitrary training inputs x_l and x_k, with $l,k=1,\ldots,N$, we get from Equation (47)

$$\begin{aligned}E_{D\sim D_0}[\alpha] &= \mathbf{K}^{-1}\mathbf{m}\\ E_{D\sim D_0}[\alpha\alpha^T]-E_{D\sim D_0}[\alpha]E_{D\sim D_0}[\alpha]^T &= \mathbf{K}^{-1}(\mathbf{Q}-\mathbf{m}\mathbf{m}^T)\mathbf{K}^{-1}\end{aligned}$$

As shown in Section 4.5, the TAP approach computes approximate values for the vector $\mathbf{m}$ and the matrix $\mathbf{Q}$. With Equations (34), (41), our final results for bootstrap mean and variance are given by

$$\begin{aligned}E_{D\sim D_0}\left[\hat{f}_x(D)\right] &= \mathbf{k}(x)^T\mathbf{T}\gamma \qquad (48)\\ E_{D\sim D_0}\left[(\hat{f}_x(D))^2\right]-\left(E_{D\sim D_0}\left[\hat{f}_x(D)\right]\right)^2 &= -\mathbf{k}(x)^T\mathbf{T}\mathrm{diag}(\lambda)\mathbf{T}^T\mathbf{k}(x)\end{aligned}$$

with $\mathbf{k}(x)=(K(x,x_1),\ldots,K(x,x_N))^T$ and $\mathbf{T}=(\mathbf{I}+\mathrm{diag}(\Delta\lambda)\mathbf{K})^{-1}$. The parameters $\gamma(i)$, $\Delta\lambda(i)$, and $\lambda(i)$ are determined together with the parameters $\gamma_c(i)$, $\Delta\lambda_c(i)$, and $\lambda_c(i)$ of the N approximate cavity distributions Equation (21). Note that Equations (48) are valid for *arbitrary* inputs x.

4.7.2 BOOTSTRAPPING THE FULL DISTRIBUTION OF THE PREDICTOR

The *marginal* distribution P_i, Equation (19), is *non-Gaussian* due to the inclusion of the local likelihood $L_i(\vec{f})$. However, it is analytically tractable for a variety of interesting "training energy" functions $h(f_i, z_i)$. Following the discussion in Section 4.6, we can compute data averages of higher moments of the predictor $\hat{f}_i(D) = \langle f_i(D) \rangle$ and generalize from this to averages of other functions g. We obtain the general result

$$E_{D\sim D_0}[g(\hat{f}_i(D))] = \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \int dG(u)\, g\left(\int df\, f\, P_i(f|u,k) \right) \tag{49}$$

where $\nu = S/N$ and g is an arbitrary function. The measure $P_i(f|u,k)$ is defined in Equation (37) and depends explicitly on the training energy $h(f_i, z_i)$. Equation (49) can be used to get a nontrivial approximation for the entire probability distribution of the estimator which is defined as $\rho_i(h) = E_{D\sim D_0}[\delta(\hat{f}_i(D) - h)]$ where $\delta(x)$ denotes the Dirac δ-distribution. For finite N and S, the exact density of the estimator at a data point i is a sum of Dirac δ-peaks. Our approximation instead yields a smoothed version of it.

5. Application to Gaussian Process Regression

The main results of the previous section are Equations (45), (46), (48) and (49). They are valid for all GP models and compute various interesting properties of the bootstrap ensemble analytically from a set of parameters provided by the TAP theory. The latter are determined by Equation (30)-(36) (which apply to *all* GP models) and Equation (38)-(40) which depend on the choice of the likelihood model Equation (10) and on the resampling scheme. In general, the set of equations can be solved iteratively. For some likelihood models, one can restrict the iteration to a specific subset of theoretical parameters.

In the following, we will consider GP regression (Neal, 1996, Williams, 1997, Williams and Rasmussen, 1996) with training energy

$$h(f_j, z_j) = \frac{1}{2\sigma^2}(f_i - y_j)^2 \,. \tag{50}$$

This model is optimally suited for a first, nontrivial test of our approximation. The estimator $\hat{f}$ of the GP regression model is obtained fairly easily and exactly without iterative methods. Note however, that we have not used the analytical formula for the estimator $\hat{f}$ in our theory. Its *explicit* form is only used for the Monte-Carlo simulation (which serves as a comparison to the theory) and is given by $\hat{f}_x(D) = \sum_{i=1}^{S} \alpha'_i K(x, x'_i)$ with $\alpha' = (\mathbf{K}' + \sigma^2 \mathbf{I})^{-1} \mathbf{y}'$ where $\mathbf{y}'$ contains all targets of the bootstrap training set D and the $S \times S$ kernel matrix $\mathbf{K}'$ is computed on the training inputs.

To complete the set of equations which determine the parameters of our theory for bootstrap averages of GP regression, we insert Equation (50) into Equations (38)-(40). This yields

$$M_i - Q_{ii} = \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \frac{1}{\Delta\lambda_c(i) + k/\sigma^2} \tag{51}$$

$$m_i = (\gamma_c(i) - y_i \Delta\lambda_c(i))(M_i - Q_{ii}) + y_i \tag{52}$$

and

$$Q_{ii} - m_i^2 = \left(\left(\frac{m_i - y_i}{M_i - Q_{ii}} \right)^2 - \lambda_c(i) \right) \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \frac{1}{(\Delta\lambda_c(i) + k/\sigma^2)^2} - (m_i - y_i)^2 \tag{53}$$

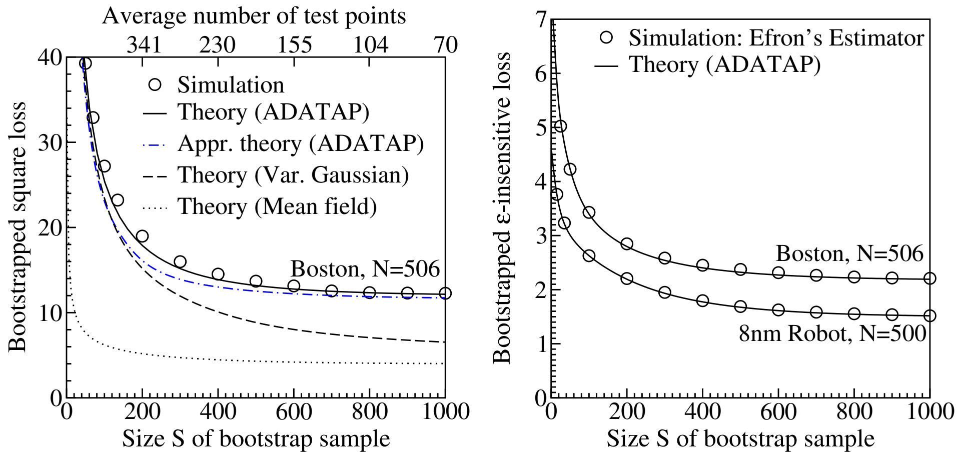

Figure 1: Bootstrapped learning curves for GP regression. *Left*: Bootstrapped square loss on Boston housing data. Comparison between simulation (circles) and 4 different approximations to the replica posterior Equation (8): ADATAP (solid line), approximate ADATAP (dot-dashed), variational Gaussian (dashed), mean field (dotted). *Right*: Bootstrapped ε-insensitive loss on Boston housing data ($N = 506$) and 8nm Robot-arm data ($N = 500$).

Close inspection of Equations (30)-(36) and Equations (51)-(53) reveals that we can solve first for $\Delta\lambda_c(i)$ and $\Delta\lambda(i)$ by iterating

$$\Delta\lambda_c(i) = (G_{ii})^{-1} - \Delta\lambda(i) \tag{54}$$

$$\Delta\lambda(i) = \left(\sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} \frac{1}{\Delta\lambda_c(i) + k/\sigma^2} \right)^{-1} - \Delta\lambda_c(i) \tag{55}$$

using the definition $\mathbf{G} = (\mathbf{K}^{-1} + \mathrm{diag}(\Delta\lambda))^{-1}$, Equation (36), where $\mathbf{K}$ denotes the $N \times N$ kernel matrix which is computed on the inputs of data set D_0. The appendix describes a method for getting typically good initial values for the iteration Equations (54) and (55) and explains how to accelerate the iteration. With $\Delta\lambda_c(i)$, $\Delta\lambda(i)$ and $\mathbf{G}$ known, all remaining parameters can be computed directly by simple matrix operations without further iterations. We obtain

$$\gamma(i) = y_i \Delta\lambda(i) \tag{56}$$

$$\lambda(i) = \sum_{j=1}^{N} (\mathbf{g} - \mathrm{diag}(\mathbf{d}))^{-1}_{ij} (m_j - y_j)^2 \tag{57}$$

where y_i denotes the target values of data set D_0, $m_i = \sum_{j=1}^N G_{ij}\gamma(j)$, $g_{ij} = (G_{ij})^2$ and the vector $\mathbf{d}$ has the entries $d(i) = \frac{H(i) g_{ii}}{H(i) - g_{ii}}$ with $H(i) = \sum_{k=0}^{\infty} \frac{\nu^k e^{-\nu}}{k!} (\Delta\lambda_c(i) + \frac{k}{\sigma^2})^{-2}$. Further

$$\gamma_c(i) = -\gamma(i) + m_i(\Delta\lambda(i) + \Delta\lambda_c(i)) \tag{58}$$

$$\lambda_c(i) = \frac{\lambda(i) g_{ii}}{H(i) - g_{ii}} + \frac{(m_i - y_i)^2}{g_{ii}} . \tag{59}$$

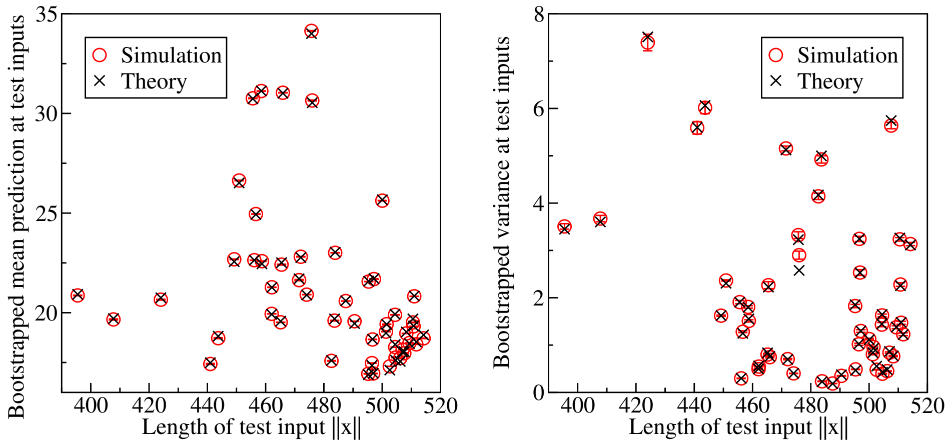

Figure 2: Bootstrapped mean (left) and variance (right) of the prediction at *test* inputs for GP regression on Boston housing data. The first 50 points of the Boston data set provide the test inputs, the remainder D_0 of the data ($N = 456$ points) was used for the bootstrap where $S = N$.

We solve Equations (54)-(59) for a given data set D_0 and covariance kernel $K(x,x')$ and plug the resulting parameters $\Delta\lambda_c(i)$, $\lambda_c(i)$ and $\gamma_c(i)$ into Equation (46). It computes the bootstrapped generalization error $\varepsilon_g(S)$ measured by an arbitrary loss function g. Figure 1 compares our theoretical predictions for the bootstrapped generalization error (solid lines) with simulation results (circles) on two benchmark data sets (boston and pumadyn-8nm, DELVE, 1996). As test measure, we have chosen square loss $g(\hat{f};x,y) = (\hat{f}_x - y)^2$ (Equation (45), Figure 1, left panel) and ε-insensitive loss (Equation (46), Figure 1, right panel)

$$g(\delta) = \begin{cases} 0 & \text{if} \quad |\delta| \in [0,(1-\beta)\varepsilon] \\ \frac{(|\delta|-(1-\beta)\varepsilon)^2}{4\beta\varepsilon} & \text{if} \quad |\delta| \in [(1-\beta)\varepsilon,(1+\beta)\varepsilon] \\ |\delta|-\varepsilon & \text{if} \quad |\delta| \in [(1+\beta)\varepsilon,\infty] \end{cases}$$

with $\delta = \hat{f}_x - y$, $\beta = 0.1$ and $\varepsilon = 0.1$. The GP model was trained with square loss Equation (50) where $\sigma^2 = 0.01$ and we used the RBF kernel $K(x,x') = \exp(-\sum_{j=1}^{d}(x_j - x'_j)^2/(v_j l^2))$ with $l^2 = 3$ for 8nm Robot-arm data (RA) and $l^2 = 73.54$ for Boston housing data (BH). v_j was set to the component-wise variance of the inputs (RA) or the square root thereof (BH). Figure 1 shows a larger part of a learning curve where the average number of *distinct* examples in the bootstrap data sets D is $S^* = N(1 - e^{-S/N})$. When the bootstrap sample size S increases, one starts to exhaust the data set D_0 which leads eventually to the observed saturation of the bootstrap learning curve. Simultaneously, one is left with a rapidly diminishing number of test points ($e^{-S/N}N$, see top axis bar of Figure 1, left panel). We observe a good agreement between theory (solid line) and simulations (circles) for the whole learning curve. For comparison, we show the learning curve (dot-dashed line) which results from a faster but approximate solution to the TAP theory. It avoids the iteration Equation (54), (55). Instead we use the start value for $\Delta\lambda$ given in Appendix A.2, compute $\mathbf{G} = (\mathbf{K}^{-1} + \mathrm{diag}(\Delta\lambda))^{-1}$, set $\Delta\lambda_c(i) = (G_{ii})^{-1} - \Delta\lambda(i)$ and solve Equations (56)-(59). The quality of this approximate solution improves with increasing sample size S. We also compare the TAP approach

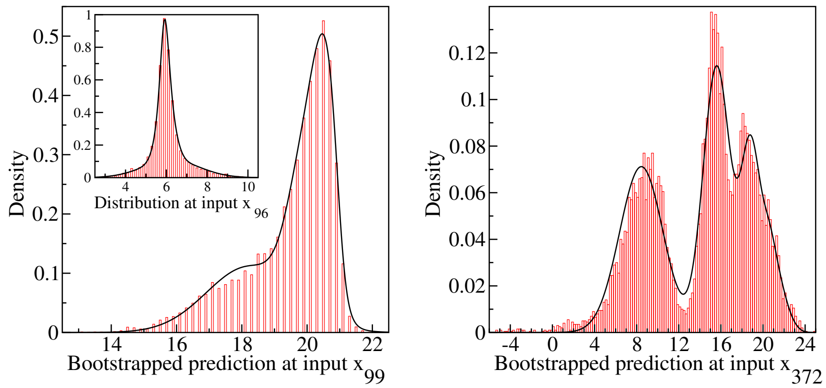

Figure 3: Bootstrapped distribution of the GP prediction $\hat{f}_i(D)$ at a given input x_i for Boston housing data, $S = N = 506$. Most distributions are unimodal with various degrees of skewness or a flank in the shoulder (left panel). The distribution may be unimodal but non-Gaussian (inset left panel) or bimodal (not shown) with a broad and a sharply concentrated component. The theory (line) describes the true distribution (histogram) in 80% of all cases very accurately and can model a high degree of structure (right panel).

with two less sophisticated approximations to the replica posterior Equation (8), the *variational Gaussian approximation* (Figure 1, left panel, dashed line) and the *mean field method* (Figure 1, left panel, dotted line). Both methods compute for integer n optimal approximations (in the Kullback-Leibler sense) to Equation (8) within a tractable family of distributions. One chooses Gaussians for the former (Malzahn and Opper, 2003) and factorizing distributions (in the example index i) for the latter approximation (see e.g., Opper and Saad, 2001). Both methods allow for a similar analytical continuation to arbitrary n as the TAP approach. We see however, that both approximations give by far less accurate results. Hence, we are not presenting the analytical formulas here.

Using Equation (56), (57) and Equation (48), we obtain analytical results for the bootstrapped mean and variance of the prediction $\hat{f}_x$ for GP regression at arbitrary inputs x. In the following, we consider the Boston housing data set which we split into a hold out set of 50 data points and a set D_0 with $N = 456$ data. Figure 2 shows results for the bootstrapped mean (left) and variance (right) of the GP prediction on the 50 *test* inputs where the bootstrap is based on resampling data set D_0 with $S = N$. We find a good agreement between our theory (crosses) and simulation results (circles). The simulation repeated the bootstrap average 5 times over sets of 5000 samples. Circles and error bars display the mean and standard deviation (square root of variance) of these 5 *average* values. Reliable numerical estimates of the bootstrapped model variance are computationally costly which emphasizes the importance of the theoretical estimate.

Finally, we can use the results on $\Delta\lambda_c(i)$, $\lambda_c(i)$ and $\gamma_c(i)$, Equations (54)-(59), to approximate the entire *distribution* of the GP prediction under the bootstrap average. The general expression Equation (49) with $g(\hat{f}_i(D)) = \delta(\hat{f}_i(D) - h)$ yields for the GP regression problem Equation (50) an *infinite mixture of Gaussians*

$$\rho_i(h) = \sum_{k=0}^{\infty} \frac{(\frac{S}{N})^k e^{-\frac{S}{N}}}{k!} \frac{(\Delta\lambda_c(i) + \frac{k}{\sigma^2})}{\sqrt{-2\pi\lambda_c(i)}} \exp\left(- \frac{\left(h\left(\Delta\lambda_c(i) + \frac{k}{\sigma^2}\right) - \gamma_c(i) - y_i \frac{k}{\sigma^2}\right)^2}{2(-\lambda_c(i))} \right) . \qquad (60)$$

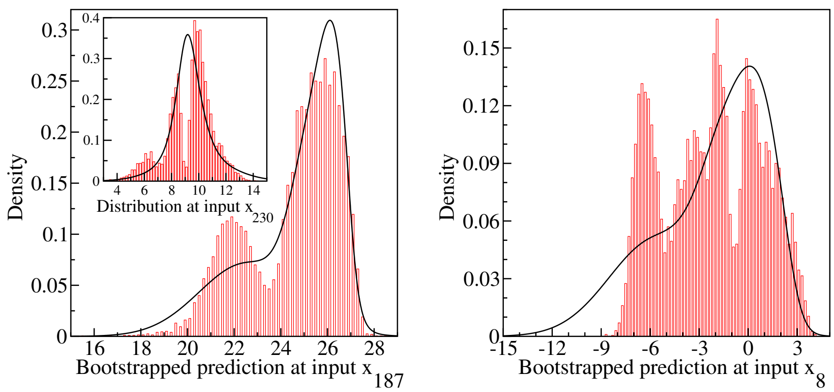

Figure 4: The left panel shows two typical examples where the theory for the bootstrap distribution (line) underestimates the amount of structure in the true distribution (histogram). The weights or the number of mixture components may be wrongly predicted (20% of all cases). The example in the right panel is very atypical (only 2% of all cases).

Figures 3 and 4 show results for the Boston housing data set where the bootstrap is based on resampling all available data ($N = 506$) with $S = N$. We computed the distributions of the GP predictions on each of the 506 inputs. Since the ADATAP approximation is based on a selfconsistent computation of first and second moments only, we should not expect that the results on the full distribution will be as accurate as the mean and variance. However, for 80% of all cases, we found that the theory (line) models the true distribution (histogram) as accurately as the examples shown in Figure 3. Most distributions are unimodal with various degrees of skewness or a shoulder in one flank (Figure 3, left). We find bimodal distributions with one broad and one sharply concentrated component (not shown). The example in the right panel of Figure 3 was selected to demonstrate that the theory can model structured densities very accurately. For 20% of all points of the data set, we found that the theory underestimates the true amount of structure in the distribution. Figure 4, left panel, shows typical examples of this effect. We found a small number of atypical cases (2%) where the theory predicts a broad unstructured distribution (Figure 4, right panel) whereas the true distribution is highly structured. The percentages above are based on optical judgment but are also well supported by similarity measures for densities. To illustrate this, we compute the bounded $L1$ distance, $L1(\rho_0, \rho) = \frac{1}{2}\int dh|\rho_0(h) - \rho(h)| \leq 1$, between the true density ρ_0 and our approximation ρ. Figure 5 shows the abundance of $L1$ values which were obtained for all 506 input points. We find $L1 \leq 0.1$ for 86.2% of all inputs and $L1 \geq 0.2$ for 2% of all inputs. The maximal value is $L1 = 0.3109$.

In contrast to other sophisticated models in machine learning, the GP regression model can be trained fairly easily by solving a set of linear equations $\mathbf{y}' = (\mathbf{K}' + \sigma^2\mathbf{I})\alpha'$ for the weights α'_i to the kernel functions $K(x, x'_i)$ (see e.g., Williams, 1997). In comparison we note that the computationally most expensive step of the ADATAP theory is the computation of the $N \times N$ matrix $\mathbf{G} = (\mathbf{K}^{-1} + \mathrm{diag}(\Delta\lambda))^{-1}$ for the iteration of Equations (54) and (55). The appendix discusses simple methods which save computation time. In both cases it suffices to compute the $N \times N$ kernel matrix $\mathbf{K}$ only once, i.e., we use cached kernel values for model training and model evaluation in the Monte-Carlo simulation. Composing data sets D_0 of various sizes N from various benchmark data, we find that

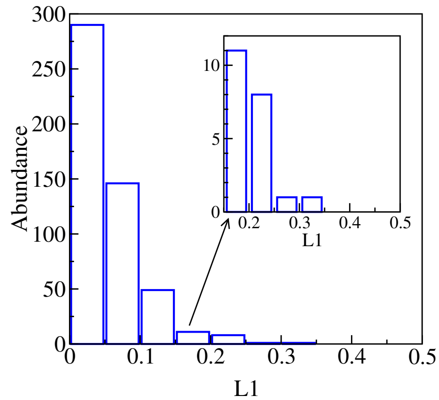

Figure 5: Histogram of $L1$-distances between the true and theoretically predicted distribution of the bootstrapped estimator on all $N = 506$ inputs of the Boston housing data set. We used histograms $\rho_0(h)$, $\rho(h)$ with bin-size $\Delta h = 0.2$ to compute $L1(\rho_0, \rho) \approx \frac{\Delta h}{2} \sum_h |\rho_0(h) - \rho(h)| \leq 1$. The inset enlarges a part of the figure.

the MATLAB program solves our theory for $S = N$ with high accuracy in the time equivalent of a Monte-Carlo average over maximal 25 samples for $N \leq 2500$ (maximal 15 samples for $N \leq 500$). Our theory is more accurate than Monte-Carlo averages with such a small amount of sampling. In the example of Figure 2 where $N = 456$, Monte-Carlo averages over 20 samples fluctuate by up to ± 0.6 (up to $\pm 3\%$) for the mean prediction and by up to ± 2.2 (up to $\pm 49\%$) for the bootstrapped variance of the GP regression model at the test points.

6. Summary and Outlook

In this paper we have presented an analytical approach to the computation of resampling averages which is based on a reformulation of the problem and a combination of the replica trick of statistical physics with an advanced approximate inference method for Bayesian models. Our method saves computational time by avoiding the multiple retraining of predictors which are usually necessary for direct sampling. It also does not require explicit analytical formulas for predictors.

So far, we have formulated our approach for GP models with general local likelihoods. Applications to a GP regression model showed promising results, where the method gives fairly accurate predictions for bootstrap test errors and for the mean and variance of GP predictions. Surprisingly, even the full bootstrap distribution is recovered well in a clear majority of cases. These results also suggest that the approximation technique used in our framework, the ADATAP method, works rather accurately compared to less sophisticated methods, like variational approximations. The non-trivial shapes of the bootstrap distributions clearly demonstrates that the ADATAP approach is not simply an approximation by a “Gaussian” but rather incorporates strongly non-Gaussian effects.

In the near future, we will give results for GP models with non-Gaussian likelihood models, like classifiers, including support-vector machines (using the well established mathematical relations between GP’s and SVM’s, see e.g., Opper and Winther, 2000). For these non-Gaussian models, training on each data sample will require to run an iterative algorithm. Hence, we expect that the computation of our approximate bootstrap (which is also based on solving a system of nonlinear equations by iteration) will have roughly the same order of computational complexity as the training on the original data set. This could give our approach a good advantage over sample based bootstrap methods, where the computational cost will scale with the number of bootstrap samples used in order to calculate averages. For a further speedup, when the number of data points is large, one may probably apply sparse approximations to kernel matrix operations, similar to those used for the training of kernel machines (see e.g., Csató and Opper, 2002, Williams and Seeger, 2001). The bootstrap estimates for classification test errors may be useful for model selection, because the expressions are not simply discrete error counts, but smooth functions of the model parameters which may be minimized more easily.[3]

While GP models seem natural candidates for an application of our new analytical approach, we view our theory as a more general framework. Hence, we will investigate if it can be applied to statistical models where model parameters are objects with a more complicated structure like trees or Markov chains. Also more sophisticated sampling schemes which could involve correlations between data points or which generate the new datasets by the trained models themselves could be of interest.

So far, an open problem remains to establish a solid rigorous foundation to the statistical physics methods used in our theory. One may hope that a further reformulation of the problem, replacing the “replica trick” by the so-called *cavity approach* (Mézard et al., 1987) can give more intuitive insights into the theory. It may also allow for the applications of recent rigorous probabilistic methods (see e.g., Talagrand) which allowed to justify previous statistical physics results obtained by the replica trick.

Acknowledgments

DM gratefully acknowledges financial support from the Copenhagen Image and Signal Processing Graduate School.

Appendix A. Analytical Bootstrap Averages for Gaussian Process Regression

This appendix explains how to solve the set of Equations (54)-(59) efficiently. They determine the values of the parameters $\Delta\lambda_c$, λ_c, γ_c and $\Delta\lambda$, λ, γ of our theory for the approximate analytical calculation of bootstrap averages for the example of Gaussian process regression.

3. The bootstrap generalization error $\varepsilon(N)$, Equation(17), estimates the bias between training error $\varepsilon_t(D_0)$ and generalization error $\varepsilon_g(D_0)$ of a learning algorithm trained on data set D_0. Take, for example, Efron's .632 estimate: $\varepsilon_g(D_0) \approx 0.368\,\varepsilon_t(D_0) + 0.632\,\varepsilon(N)$ (see also Efron and Tibshirani, 1997).

A.1 The Algorithm

Require: Data set $D_0 = \{(x_i, y_i); i = 1 \dots N\}$
Compute kernel matrix $\mathbf{K}$ on inputs of D_0.
Compute eigenvalues ω of $\mathbf{K}$.
For bootstrap sample size S:
Initialize: Find root $\Delta\lambda$ of Equation (62) with Equation (61). (Single one-dimensional root search.)
Iterate:
Update $\Delta\lambda_c$ from $\Delta\lambda$ according to Equation (54).
Update $\Delta\lambda$ from $\Delta\lambda_c$ according to Equation (55).
Until converged
Obtain γ, λ according to Equation (56), (57).
Obtain γ_c, λ_c according to Equation (58), (59).
Bootstrapped test error by Equation (46); bootstrapped distribution of estimator by Equation (60)
Bootstrapped mean prediction and variance by Equations (48)
End for

A.2 Algorithm Initialization

The algorithm solves Equation (54), (55) iteratively which requires a good initialization for the $\Delta\lambda(i)$'s. A reasonable initialization can be obtained in the following way: We neglect the dependence of $\mathbf{G}_{ii} \approx G$ and of $\Delta\lambda(i) \approx \Delta\lambda$ on the index i and write

$$G \approx \frac{1}{N}\sum_{i=1}^{N}\mathbf{G}_{ii} = \frac{1}{N}\mathrm{Tr}(\mathbf{K}^{-1} + \mathrm{diag}(\Delta\lambda))^{-1} \approx \frac{1}{N}\sum_{k=1}^{N}\frac{\omega_k}{1+\omega_k\Delta\lambda} \tag{61}$$

where ω_k for $k = 1, \dots, N$ are the eigenvalues of the kernel matrix $\mathbf{K}$. Using the same approximation within Equation (54), (55) yields

$$G = \sum_{k=0}^{\infty}\frac{\nu^k e^{-\nu}}{k!}\,\frac{G}{1 - G\,(\Delta\lambda(i) - k/\sigma^2)}\ . \tag{62}$$

Solving Equations (61) and (62) with respect to $\Delta\lambda$ by a *one dimensional* root finding routine gives the initialization for the iteration of Equations (54) and (55). The iteration is found to be stable and shows fast convergence whereby the number of required iterations decreases with increasing sample size S.

For large N, one can save time by computing Equation (61) with the eigenvalues ω of a smaller kernel matrix based on a random subset of $\frac{N}{P}$ of the data (replace $\frac{1}{N}$ by $\frac{P}{N}$ in Equation (61)). The choice $P = 4$ yields start values for $\Delta\lambda$ which are slightly degraded but equally efficient for the iteration.

A.3 Standard Iteration Step

The t-th iteration uses $\Delta\lambda^t$ to compute the matrix $\mathbf{G}^t = (\mathbf{K}^{-1} + \mathrm{diag}(\Delta\lambda^t))^{-1}$. This is the most time consuming step of the TAP theory. We remark that we can easily rewrite $\mathbf{G}^t$ to avoid computation of $\mathbf{K}^{-1}$ (which may be close to singular). Under MATLAB it pays off to use the division operator on *symmetric* matrices

$$\mathbf{G}^t = \mathrm{diag}(\Delta\lambda^t)^{-1}\left(\mathrm{diag}(\Delta\lambda^t)^{-1} + \mathbf{K}\right)^{-1}\mathbf{K} \tag{63}$$

From G^t_{ii} and Equation (54), (55) we obtain the updates $\Delta\lambda_c{}^{t+1}$, $\Delta\lambda^{t+1}$. The solution has usually the property that $\Delta\lambda_c(i) >> \Delta\lambda(i)$ where $\Delta\lambda_c(i)$ increases significantly with S. We determine $\Delta\lambda_c(i)$, $\Delta\lambda(i)$ with at least $\pm 10^{-3}$ accuracy relative to their absolute values. We define the absolute error at iteration step t by

$$\delta = \Delta\lambda^{t+1} - \Delta\lambda^t \; . \tag{64}$$

The number A of sites with changes $|\delta_j| > 10^{-4}$ drops to values $A << N$ after typically $2-3$ iterations. We store the corresponding site indices in an A dimensional vector $\mathcal{J}(A)$. For $A << N$, we can compute the matrix update Equation (63) more efficiently using the Woodbury formula

$$\mathbf{G}^{t+1} = \mathbf{G}^t - \mathbf{U}\,(\mathbf{I}+\mathbf{W})^{-1}\,\mathbf{U}^T \; . \tag{65}$$

$\mathbf{U}$ is $N \times A$ dimensional with entries $G^t_{ik}\sqrt{\delta_k}$ where $i = 1,\dots,N$ and $k \in \mathcal{J}(A)$, i.e., the columns of $\mathbf{U}$ are proportional to the A columns of G^t which correspond to active sites $\mathcal{J}(A)$. The identity matrix $\mathbf{I}$ and matrix $\mathbf{W}$ are both $A \times A$ dimensional. $\mathbf{W}$ has the entries $G^t_{k_1,k_2}\sqrt{\delta_{k_1}\delta_{k_2}}$ with $k_1, k_2 \in \mathcal{J}(A)$.

A.4 Approximate Iteration Step

The iteration requires only updates of the diagonal elements G^{t+1}_{ii} (see Equation (54)). This subsection discusses an *approximate* update for G^{t+1}_{ii} which saves time and aids convergence to small active sets A. We will place such approximate updates between *exact* updates of $\mathbf{G}^{t+1}$ by Equation (63) or (65). The latter ensures that we do not accumulate errors.

We regard $\mathbf{G}^t$ as an approximation to the unknown matrix $\mathbf{G}^{t+1}$, define the approximation error by $\mathbf{R} \doteq \mathbf{I} - \mathbf{G}^t(\mathbf{G}^{t+1})^{-1} = -\mathbf{G}^t\mathrm{diag}(\delta)$ and get (Press et al., 1992)

$$\mathbf{G}^{t+1} = (\mathbf{I}-\mathbf{R})^{-1}\mathbf{G}^t = \left(\mathbf{I} + \sum_{l=1}^{\infty}\mathbf{R}^l\right)\mathbf{G}^t \tag{66}$$

which is determined by the changes δ, Equation (64), and by $\mathbf{G}^t$. $\mathbf{G}^t$ has typically small entries $G^t_{ij} << 1$ (for $K_{ij} \le 1$).[4] Equation (66) enables us to obtain *approximate* updates for G^{t+1}_{ii} which require only $O(N^2)$ operations. With $g_{ij} = (G^t_{ij})^2$ we approximate

$$G^{t+1}_{ii} \approx G^t_{ii} - \sum_{j=1}^{N} g_{ij}\left(\delta_j - \delta_j^2 G^t_{jj} + \delta_j^3 g_{jj}\right) \tag{67}$$

Equation (67) is non-local with respect to δ. The quadratic and cubic terms in δ_j approximate the second and third order contributions $(R^2 + R^3)G^t$ under the assumption that off-diagonal entries in $\mathbf{G}^t$ are small in comparison to diagonal entries. Note that Equation (67) uses the values $G^t_{ll} = G^t_{ll}(\Delta\lambda^t)$ from our last *exact* computation with $\Delta\lambda^t$. We define $\bar{G}^t = \frac{1}{N}\sum_{l=1}^{N} G^t_{ll}$ and find that if $|\delta_j|\bar{G}^t < 0.1$ for all sites $j = 1,\dots,N$, we can do repeated iterations using Equation (67) where we update $\Delta\lambda^{t+1}$ (and $\Delta\lambda_c{}^{t+1}$) but keep G^t_{ll}, $\Delta\lambda^t$ unchanged. δ is updated according to Equation (64). It is beneficial to do up to 3 iterations before recomputing $\mathbf{G}^{t+1}$ from the final $\Delta\lambda^{t+1}$ by an exact method.

4. The values G_{ij} decrease with increasing sample size S.

References

L. Breiman. Bagging Predictors. *Machine Learning*, 24:123-140, 1996.

L. Csató and M. Opper. Sparse Gaussian processes. *Neural Computation*, 14(3): 641 - 668, 2002.

L. Csató, M. Opper and O. Winther. TAP Gibbs free energy, belief propagation and sparsity. *Advances in Neural Information Processing Systems*, 14, T. G. Dietterich, S. Becker and Z. Ghahramani, editors. MIT Press, Cambridge, MA, 2002.

Delve: Data for Evaluating Learning in Valid Experiments. Copyright (c) 1995-1996 by The University of Toronto, Toronto, Ontario, Canada. [http://www.cs.toronto.edu/~delve/].

B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7: 1-26, 1979.

B. Efron. The Jackknife, the Bootstrap and Other Resampling Plans. *CBM-NSF Regional Conference Series in Applied Mathematics*, 1982.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability 57, Chapman & Hall, 1993.

B. Efron and R. J. Tibshirani. Improvements on cross-validation: The 632+bootstrap method. *J. Amer. Statist. Assoc.* 92: 548-560, 1997.

D. Malzahn and M. Opper. A statistical mechanics approach to approximate analytical bootstrap averages. *Advances in Neural Information Processing Systems* 15: 327-334, S. Becker, S.Thrun and K. Obermayer, editors. MIT Press, Cambridge, MA, 2003.

M. Mézard, 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 118, Springer, 1996.

M. Opper and D. Saad, editors. *Advanced Mean Field Methods: Theory and Practice*. MIT Press, 2001.

M. Opper and O. Winther. Gaussian processes for classification: Mean field algorithms. *Neural Computation*, 12: 2655-2684, 2000.

M. Opper and O. Winther. Tractable approximations for probabilistic models: The adaptive TAP approach. *Phys. Rev. Lett.* , 86: 3695, 2001.

M. Opper and O. Winther. Adaptive and self-averaging Thouless-Anderson-Palmer mean field theory for probabilistic modeling. *Phys. Rev. E*, 64, 056131, 2001.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Second Edition, 1992.

B. Schölkopf, C. J. C. Burges, A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

J. Shao and D. Tu. *The Jackknife and Bootstrap*. Springer Series in Statistics, Springer, 1995.

M. Talagrand. Many results and references can be found on M. Talagrand's webpage: `http://www.proba.jussieu.fr/users/talagran/ index.html`

C. K. I. Williams. Regression with Gaussian Processes. *Mathematics of Neural Networks: Models, Algorithms and Applications*, S. W. Ellacott, J. C. Mason and I. J. Anderson, editors. Kluwer, 1997.

C. K. I. Williams and C. E. Rasmussen. Gaussian Processes for Regression. *Advances in Neural Information Processing Systems* 8: 514, D. S. Touretzky, M. C. Mozer and M. E. Hasselmo, editors. MIT Press, Cambridge, MA, 1996.

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems* 13: 682-688, T. K. Leen, T. G. Diettrich and V. Tresp, editors. MIT Press, Cambridge, MA, 2001.