Transductive and Inductive Methods for Approximate Gaussian Process Regression

Anton Schwaighofer and Volker Tresp
Siemens Corporate Technology, Department of Neural Computation
Otto-Hahn-Ring 6, 81739 Munich, Germany

Abstract

Gaussian process regression allows a simple analytical treatment of exact Bayesian inference and has been found to provide good performance, yet scales badly with the number of training data. In this paper we compare experimentally three of the leading approaches towards scaling Gaussian processes regression to large data sets: the subset of representers method, the reduced rank approximation, and the Bayesian committee machine. Furthermore we provide theoretical insight into some of our experimental results. We found that subset of representers methods can give good and particularly fast predictions for data sets with high and medium noise levels. On low noise data sets, the Bayesian committee machine achieves significantly better accuracy, yet at a higher computational cost for large test data sets.

1 Introduction

Gaussian process regression (GPR) has demonstrated excellent performance in a number of applications. One unpleasant aspect of GPR is its scaling behavior with the size of the training data set N. In direct implementations, training time increases as $O(N^3)$ whereas memory footprint and prediction time are proportional to training data size. The subset of representer method (SRM), the reduced rank approximation (RRA) and the Bayesian committee machine (BCM) are three related approaches which solve the scaling problems based on a finite dimensional approximation to the typically infinite dimensional Gaussian process.

The focus of this paper is on providing a unifying view on the methods and analyze their differences, both from an experimental and a theoretical point of view. A major difference of the methods discussed here is that the BCM performs *transduction*, i.e. it exploits knowledge about the location of the test data in its approximation. As a consequence, the BCM approximation is calculated when the inputs to the test data are known. On the other hand, RRA and SRM methods perform *induction* style learning, which means that the model parameters are calculated solely based on the training data. In this paper we complete the picture and also formulate induction for the BCM and transduction both for RRA and SRM methods. We examine asymptotical and actual runtime of the different approaches and investigate the accuracy versus speed trade-off. In Sec. 2 we will briefly introduce Gaussian process regression (GPR). Sec. 3 presents the various approaches to scaling GPR to large data sets. Sec. 4 follows with an experimental comparison of the described approaches. In Sec. 5 we analyze the experimental results and discuss the concepts of induction and transduction, followed by Sec. 6 with conclusions.

2 Gaussian Processes

In a Bayesian treatment of Gaussian process regression (GPR), one assumes that data are generated based on an unknown function f, where a Gaussian prior distribution $P(f)$ over the space of functions is assumed. As a consequence of the Gaussian assumption, *a priori* functional values $f(\mathbf{x}_i)$ on points $\{\mathbf{x}_i\}_{i=1}^N$ are jointly Gaussian distributed, with zero mean and covariance matrix K^N. The covariance matrix (or Gram matrix) K^N itself is given by the kernel (or covariance) function $k(\cdot,\cdot)$, with $K_{ij}^N = k(\mathbf{x}_i, \mathbf{x}_j)$.

For GPR, we assume a set of training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where targets are generated from f via $y_i = f(\mathbf{x}_i) + e_i$. Here, e_i is independent additive Gaussian noise with variance σ^2. We denote the vector of observations y_i by $\mathbf{y} = (y_1, \ldots, y_N)^\top$. The Bayes optimal estimator $\hat{f}(x) = E(f(x)|\mathcal{D})$ takes on the form of a weighted combination of kernel functions [4] on training points $\mathbf{x}_i$

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} w_i k(\mathbf{x}, \mathbf{x}_i). \tag{1}$$

The weight vector $\mathbf{w} = (w_1, \ldots, w_N)^\top$ is the solution to the system of linear equations

$$(K^N + \sigma^2 \mathbf{1})\mathbf{w} = \mathbf{y} \tag{2}$$

where $\mathbf{1}$ denotes a unit matrix. Mean and covariance of the GP prediction $\mathbf{f}^*$ on a set of test points $\mathbf{x}_1^*, \ldots, \mathbf{x}_T^*$ can be written conveniently as

$$E(\mathbf{f}^*|\mathcal{D}) = K^{*N}\mathbf{w} \text{ and } \operatorname{cov}(\mathbf{f}^*|\mathcal{D}) = K^* - K^{*N}(K^N + \sigma^2\mathbf{1})^{-1}(K^{*N})^\top. \tag{3}$$

with $K_{ij}^{*N} = k(\mathbf{x}_i^*, \mathbf{x}_j)$. Eq. (2) shows clearly what problem we may expect with large training data sets: The solution to a system of N linear equations requires $O(N^3)$ operations, and the size of the Gram matrix K^N may easily exceed the memory capacity of an average work station.

3 Approximation Methods for GPR

3.1 Bayesian Committee Machine (BCM)

In the BCM [8], the training data $\mathcal{D}$ are partitioned into M disjoint sets $\mathcal{D}^1, \ldots, \mathcal{D}^M$ of approximately same size ("modules"), and M learning systems are trained on their respective training data set. The BCM calculates the unknown responses at a number of test points $\mathbf{x}_1^* \ldots \mathbf{x}_T^*$ at the same time. Let $\mathbf{f}^* = (f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_T^*)$ be the vector of these response variables at the T test points. The underlying assumption of the BCM is that

$$P(\mathcal{D}^i|\mathbf{f}^*, \mathcal{D} \setminus \mathcal{D}^i) \approx P(\mathcal{D}^i|\mathbf{f}^*) \quad i = 1, \ldots, M$$

meaning that individual parts $\mathcal{D}^i$ of the data are independent given $\mathbf{f}^*$. This is a good assumption if (1) the test set contains many points, since these points are sufficient to define the map f and make all data independent, or (2) if each portion of data $\mathcal{D}^i$ is large, which increases their independence from each other on average, or (3) if each $\mathcal{D}^i$ contains data that is spatially separated from other training data, since spatially separated data points tend to be independent even unconditionally.

Under the above independence assumption, one obtains an estimate of $\mathbf{f}^*$ at the test data as

$$\hat{E}(\mathbf{f}^*|\mathcal{D}) = C^{-1} \sum_{i=1}^{M} \operatorname{cov}(\mathbf{f}^*|\mathcal{D}^i)^{-1} E(\mathbf{f}^*|\mathcal{D}^i) \tag{4}$$

$$\text{with} \quad C = \widehat{\operatorname{cov}}(\mathbf{f}^*|\mathcal{D})^{-1} = -(M-1)(K^*)^{-1} + \sum_{i=1}^{M} \operatorname{cov}(\mathbf{f}^*|\mathcal{D}^i)^{-1}. \tag{5}$$

Here, K^* is the $T \times T$ prior covariance matrix at the test points. $E(\mathbf{f}^*|\mathcal{D}^i)$ and $\mathrm{cov}(\mathbf{f}^*|\mathcal{D}^i)$ are obtained from Gaussian process regression on module $\mathcal{D}^i$ via Eq. (3).

Eq. (4) has the form of a committee machine where the predictions of the committee members at all T test points are used to form the prediction of the committee at those points. The prediction of each module i is weighted by the inverse covariance of its prediction. An intuitively appealing effect of this weighting scheme is that modules which are uncertain about their predictions are automatically weighted less than modules that are certain about their predictions. Note that the BCM is a transductive method, in that it only can be applied once the test patterns are known. To emphasize this, we will refer this method as BCM Trans.

In previous work on the BCM, data was partitioned randomly into modules $\mathcal{D}^i$. In this paper we report for the first time results where data were assigned via k-means clustering, thereby improving the independence assumption of Eq. (3.1), see the above remark (3). Later on, we will refer to this method as BCM TransClust.

We may also use the BCM for induction style learning (BCM Ind) by, in the training stage, inferring the functional values $\mathbf{f}^B$ for a set of B basis points $\{\mathbf{x}_i^B\}$ chosen out of the training data $\mathcal{D}$. In the test stage, one can project from basis points to test points $\{\mathbf{x}_i^*\}_{i=1}^T$ using Eq. (7.1) of [8] and obtain for the functional values $\mathbf{f}^*$ on the test points

$$\hat{E}(\mathbf{f}^*|\mathcal{D}) = K^{*B}(K^B)^{-1}\hat{E}(\mathbf{f}^B|\mathcal{D}) \tag{6}$$

with $K_{ij}^{*B} = k(\mathbf{x}_i^*, \mathbf{x}_j^B)$ and $K_{ij}^B = k(\mathbf{x}_i^B, \mathbf{x}_j^B)$.

3.2 Reduced Rank Approximation (RRA)

Reduced rank approximations focus on ways of efficiently solving the system of linear equations Eq. (2), by replacing the kernel matrix K^N with some suitable approximation $\tilde{K}^N$. One suitable form for $\tilde{K}^N$ can be obtained from a truncated eigendecomposition

$$K^N \approx \tilde{K}^N = U\Lambda U^\top$$

where U is an $N \times B$ matrix containing B eigenvectors of K^N and Λ is a diagonal matrix with B eigenvalues.

Williams and Seeger [10] use the Nyström method to calculate an approximation to the first B eigenvalues and eigenvectors of K^N. Essentially, the Nyström method performs an eigen decomposition of the $B \times B$ covariance matrix K^B, obtained from a set of B basis points selected at random out of the training data. Based on this decomposition, B eigenvectors and eigenvalues of K^N are estimated. Using the matrix inversion lemma, one obtains an approximate solution for the weight vector $\mathbf{w}$ in Eq. (1) as

$$\mathbf{w} \approx \frac{1}{\sigma^2}\left(\mathbf{1} - K^{NB}\left[(K^{NB})^\top K^{NB} + \sigma^2(K^B)^{-1}\right]^{-1}(K^{NB})^\top\right)\mathbf{y}.$$

We refer to this method as RRA Nyst. Here, K^{NB} is the kernel matrix of training points versus base points. In this form, only matrices of size $B \times B$ need to be inverted, with usually $B \ll N$, instead of the $N \times N$ matrices in the direct computation. Mind that the decomposition of K^N is only used to obtain an efficient way of solving the linear system Eq. (2), the covariance of the Gaussian process is left unchanged. The prediction obtained with the Nyström method still has the form of a superposition of all N kernel functions, as given in Eq. (1).

3.3 Subset of Representers Method (SRM)

The starting point for the SRM is a decomposition of the covariance function of the form

$$k(\mathbf{x}_i, \mathbf{x}_j) \approx K^{i,B}(K^B)^{-1}(K^{j,B})^\top.$$

Here, K^B is the Gram matrix for a subset of B basis points and $K^{i,B}$ is the vector of covariances between $\mathbf{x}_i$ and the basis points. It is well known that, if such a decomposition exists, a Gaussian process is equivalent to a system with fixed basis functions, where the fixed basis functions are given by kernels at the training points $k(\cdot, \mathbf{x}_i)$. For such a system of fixed basis functions, $(K^B)^{-1}$ is the covariance of a Gaussian prior distribution for the weights on the basis functions (see, for example, [8]). In other words, by using the approximation Eq. (3.3) we have transformed the original infinite dimensional Gaussian process into a system with B fixed basis functions, where predictions can be calculated as

$$\tilde{f}(\mathbf{x}) = \sum_{i=1}^{B} \beta_i k(\mathbf{x}, \mathbf{x}_i)$$

with an optimal weight vector

$$\beta = (\sigma^2 K^B + (K^{NB})^\top K^{NB})^{-1} (K^{NB})^\top \mathbf{y}. \tag{7}$$

In practical implementation, one may expect different performance depending on the choice of the B basis points $\mathbf{x}_1, \ldots, \mathbf{x}_B$. Different approaches for basis selection have been used in literature, we will discuss them in turn.

Obviously, one may select the basis points at random (SRM Rand) out of the training set. While this produces no computational overhead, the prediction outcome may be suboptimal.

In the sparse greedy matrix approximation (SRM SGMA, [6]) a subset of B basis kernel functions is selected such that all kernel functions on the training data can be well approximated by linear combinations of the selected basis kernels. If proximity in the associated reproducing kernel Hilbert space (RKHS) is chosen as the approximation criterion, one obtains exactly the form of kernel function given in Eq. (3.3). Smola and Schölkopf [6] introduce a greedy algorithm that finds a near optimal set of basis functions, where the algorithm has the same (asymptotic) computational complexity $O(NB^2)$ as the SRM Rand method.

Whereas the SGMA basis selection desribed above focuses only on the representation power of kernel functions, one can also design a basis selection scheme that takes into account the full likelihood model of the Gaussian process. The underlying idea of the greedy posterior approximation algorithm SRM GP [7] is to compare the log posterior of the subset of representers method and the full Gaussian process log posterior. One thus can select basis functions in such a fashion that the SRM log posterior best approximates[1] the full GP log posterior, while keeping the total number of basis functions B minimal. As for the case of SGMA, this algorithm can be formulated such that its asymptotical computational complexity is $O(NB^2)$, where B is the total number of basis functions selected.

Finally, one can devise a transductive method (SRM Trans) by using the test points as basis points. Mind that for transduction, the SRM and the RRA Nyst methods give identical predictions.

3.4 Computational Cost

Table 1 shows the asymptotic computational cost for all approximation methods we have described in Sec. 3.1 through 3.3. The subset of representers methods (SRM) show the most favorable cost for the prediction stage, since the resulting model consists only of B basis functions with associated weight vector. Note that the $O(\cdot)$ notation is hiding constant factors, therefore methods with the same asymptotical complexity may exhibit significantly different time consumption in practice.

[1] However, Rasmussen [5] noted that Smola and Bartlett [7] falsely assume that the additive constant terms in the log likelihood remain constant during basis selection.

Method	Memory consumption		Computational cost	
	Initialization	Prediction	Initialization	Prediction
Exact GPR	$O(N^2)$	$O(N)$	$O(N^3)$	$O(N)$
BCM Trans	—	$O(N+B^2)$	—	$O(NB)$
BCM Ind	$O(N+B^2)$	$O(B)$	$O(NB^2)$	$O(B)$
SRM Rand, SRM SGMA, SRM GP, SRM Trans	$O(NB)$	$O(B)$	$O(NB^2)$	$O(B)$
RRA Nyst	$O(NB)$	$O(N)$	$O(NB^2)$	$O(N)$

Table 1: Asymptotic computational cost and memory consumption different GP approximation methods with N training data points and B basis points, $B \ll N$. For the BCM and its variants, we assume here that training and test data are partitioned into modules of size B. All costs for predictions show the cost per test point.

4 Experimental Comparison

In this section we will present a comparison of the different approximation methods discussed in Sec. 3. In the ABALONE data set [1] with 4177 examples, the goal is to predict the age of Abalones based on 8 inputs. The KIN8NM data set[2] represents the forward dynamics of an 8 link all-revolute robot arm, based on 8192 examples. The goal is to predict the distance of the end-effector from a target, given the twist angles of the 8 links as features. KIN40K represents the same task, yet has a low noise level (as compared to KIN8NM with medium noise level) and contains 40.000 examples. Data set ART with 50000 examples was used extensively in [8] and describes a nonlinear map with 5 inputs with a small amount of Gaussian additive noise.

For all data sets, we used a squared exponential kernel of the form $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2d^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$, where the kernel parameter d was optimized individually for each method. To allow a fair comparison, the subset selection methods SRM SGMA and SRM GP were forced to select a given number B of basis functions (instead of using the stopping criteria proposed by the authors of the respective methods). Thus, all methods form their predictions as a linear combination of exactly B basis functions.

Table 2 shows the average variance explained[3] in a 10-fold cross validation procedure on all data sets. For each of the methods, we have run experiments with different kernel width d, in Table 2 we list only the results obtained with optimal d for each method.

On the ABALONE data set (very high level of noise), all of the tested methods achieved almost identical performance. For all other data sets, significant performance differences were observed. Out of the inductive methods (SRM SGMA, SRM Rand, SRM GP, RRA Nyst, BCM Ind) best performance was always achieved with SRM GP. Using the results in a paired t-test showed that this was significant at a level of 99% or above. The inductive BCM Ind can not be recommended in its current form, since its performance is inferior to other inductive methods. Furthermore, we observed certain problems with the RRA Nyst method. On all but the ABALONE data set, weights $\mathbf{w}$ took on values in the range of 10^3 or above, leading to poor performance. Details on that observation will be given in Sec. 5.2.

Comparing induction and transduction methods, we see that the transductive BCM performs significantly better in most cases. In particular, the BCM TransClust method with pre-clustered training data, achieved very good performance on the low noise data sets KIN40K and ART.

[2] From the DELVE archive `http://www.cs.toronto.edu/~delve/`

[3] variance explained $= 1 - \frac{MSE_{\text{model}}}{MSE_{\text{mean}}}$, where MSE_{mean} is the MSE obtained from using the mean of training data mean as a constant predictor. This gives a measure of performance that is independent of data scaling.

	Abalone		KIN8NM		KIN40K		ART	
Method	200	1000	200	1000	200	1000	200	1000
SRM GP	57.19	57.19	86.21	**92.16**	90.51	97.64	96.09	98.88
SRM SGMA	57.17	57.19	78.16	91.30	81.68	95.75	94.38	98.21
SRM Random	57.14	57.18	77.66	90.99	81.23	95.61	94.13	98.21
BCM Ind	57.21	57.19	73.08	90.34	75.73	94.83	93.62	97.98
RRA Nyst	57.02	56.90	N/A	N/A	N/A	N/A	N/A	N/A
BCM TransClust	57.14	57.19	**89.68**	91.69	**97.19**	**99.17**	**99.73**	**99.80**
BCM Trans	57.13	57.20	86.96	91.37	91.22	97.89	97.31	99.09
SRM Trans	57.03	57.21	78.05	90.21	83.53	95.75	94.85	98.36

Table 2: Variance explained, obtained with different GPR approximation methods on four data sets, with different number of basis functions selected (200 or 1000). Variance explained is given in per cent, averaged over 10-fold cross validation. Marked in bold are results that are significantly better (with a significance level of 99% or above in a paired t-test) than any of the other methods

Here, the average MSE obtained with BCM TransClust was only a fraction (25-30%) of the average MSE of the best inductive method. By a paired t-test we confirmed that the BCM TransClust method is significantly better than all other methods on the KIN40K and ART data sets, with significance level of 99% or above. On the KIN8NM data set (medium noise level) we observed a case where SRM GP performed best. We attribute this to the fact that k-means clustering was not able to find good spatially separated clusters, thus weakening the independence assumptions in the BCM. We further noticed that, on the KIN40K and ART data sets, SRM Trans consistently outperformed SRM Random, despite of SRM Trans being the most simplistic tranductive method.

As mentioned above, we did not make use of the stopping criterion proposed by Smola and Bartlett [7] for the SRM GP method, namely the relative gap between SRM log posterior and the log posterior of the full Gaussian process model. In Smola and Bartlett [7], the authors suggest that the gap is indicative of the generalization performance of the SRM model and use a gap of 2.5% in their experiments. In contrast, we did not observe any correlation between the gap and the generalization performance in our experiments. For example, selecting 200 basis points out of the KIN40K data set gave a gap of $\approx 1\%$, indicating a good fit. As shown in Table 2, a significantly better error was achieved with 1000 basis functions (giving a gap of $\approx 3.5 \cdot 10^{-4}$). Thus, the question of choosing an appropriate basis set size B remains open.

It is also interesting to consider not only the asymptotic complexity, but the actual runtime as well. For one (out of 10) cross validation runs on KIN40K (36000 training examples, 4000 test patterns, $B = 1000$ basis functions), SRM Rand and RRA Nyst took about 3 minutes, the BCM methods on the order of 30 minutes, SRM SGMA on the order of 6 hours, and SRM GP took about 10 hours. Of course, all SRM methods have significant advantages in terms of runtime once the basis set is selected. We thus consider the time spent for basis selection as the bottleneck for SRM methods with larger number of basis functions.

5 Discussion of the Results

5.1 Transduction versus Induction

Our experimental results have shown that, except for one case, transduction using the BCM gives better results than any of the inductive methods. We can gain insight by considering the decomposition $P(\mathbf{f}^*, \mathcal{D}) = P(\mathbf{f}^*)P(\mathcal{D}|\mathbf{f}^*)$, where $P(\mathbf{f}^*)$ is the Gaussian process prior for functional values at the test points. Based on this decomposition the GPR prediction Eq. (3) can now be

written as

$$E(\mathbf{f}^*|\mathcal{D}) = K^* \left(K^* + K^{*N} \operatorname{cov}(\mathbf{y}|\mathbf{f}^*)^{-1} (K^{*N})^\top \right)^{-1} K^{*N} \operatorname{cov}(\mathbf{y}|\mathbf{f}^*)^{-1} \mathbf{y} \tag{8}$$

$$\text{where} \quad \operatorname{cov}(\mathbf{y}|\mathbf{f}^*) = K^N + \sigma^2 \mathbf{1} - (K^{*N})^\top (K^*)^{-1} K^{*N} \tag{9}$$

Note, that in Eq. (8), data are weighted by the inverse covariance of the Gaussian distribution $\operatorname{cov}(\mathbf{y}|\mathbf{f}^*)$ of the data given the functional values at the test points. Eq. (9) reveals that those data points are weighted less (i.e. have a higher effective variance), which cannot be predicted well from the functional values of the test points. As a result, data points which are in this sense closer to the test points (in that they can be predicted better) obtain a higher weight than data which are remote from the test points. By comparing Eq. (7) and Eq. (8) one can see that the latter is identical to SRM Trans, if we set $\operatorname{cov}(\mathbf{y}|\mathbf{f}^*) = \sigma^2 \mathbf{1}$. Thus, transduction with SRM performs an approximation by replacing the proper weighting by a constant weighting on all training data.

On the other hand, it can be shown that the BCM Trans performs a block diagonal approximation of $\operatorname{cov}(\mathbf{y}|\mathbf{f}^*)$, thus computing a test set dependent weighting of training data. The block diagonal approximation works well if blocks are large (i.e. module size is large) and if the number of test points T is large since then $\operatorname{cov}(\mathbf{y}|\mathbf{f}^*)$ tends to be diagonal. It is also apparent that data partitioning through clustering is beneficial. Although one is actually interested in arranging $\operatorname{cov}(\mathbf{y}|\mathbf{f}^*)$ to be block diagonal, a similar effect can be achieved by clustering the training data, leading to a block diagonal K^N. In the relevant case where data points can not well predicted from f^*, the two remaining terms in Eq. (9) cancel and a block diagonal K^N results in a block diagonal $\operatorname{cov}(\mathbf{y}|\mathbf{f}^*)$. Summing up, transduction in the form of the BCM works, since the solution weights training data that is close to test data more heavily. A drawback is of course that this is a test data dependent approximation, in that it needs to be re-calculated each time new test data arrive.

5.2 Problems with the RRA

As mentioned in Sec. 4, we observed that weights $\tilde{\mathbf{w}}$ tend to have unreasonably large values, sometimes in the range of 10^3 or above, on data sets KIN8NM, KIN40K and ART. We might explain that by considering the perturbation of linear systems. RRA Nyst solves Eq. (2) with an approximate $\tilde{K}^N$ instead of K^N, thus calculating an approximate $\tilde{\mathbf{w}}$ instead of the true $\mathbf{w}$. A result from matrix perturbation theory states that the relative error of the approximate $\tilde{\mathbf{w}}$ is bounded by

$$\frac{\|\tilde{\mathbf{w}} - \mathbf{w}\|}{\|\mathbf{w}\|} \leq \|(\tilde{K}^N + \sigma^2 \mathbf{1})^{-1} E\| \tag{10}$$

with perturbation matrix $E = (K^N + \sigma^2 \mathbf{1}) - (\tilde{K}^N + \sigma^2 \mathbf{1})$. In the above equation we have used $\|\cdot\|$ for both matrix and vector norm, where—for the bound to hold—the two norms must be consistent (see for example [2], chapter 5).

Consistent norms are for example the Euclidean norm for vectors and the spectral norm for matrices. Using these norms, we can write the above bound as

$$\frac{\|\tilde{\mathbf{w}} - \mathbf{w}\|}{\|\mathbf{w}\|} \leq \max_i \frac{|\lambda_i - \tilde{\lambda}_i|}{\tilde{\lambda}_i + \sigma^2}$$

where λ_i and $\tilde{\lambda}_i$ denote eigenvalues of K^N resp. $\tilde{K}^N$.

By its nature, the Nyström approximation is able to estimate at most B eigenvalues of the Gram matrix K, where B is the number of basis points. In an optimistic setting we may assume that the largest B eigenvalues have been correctly estimated and that eigen values are sorted by magnitude. Then

$$\frac{\|\tilde{\mathbf{w}} - \mathbf{w}\|}{\|\mathbf{w}\|} \leq \frac{\lambda_{B+1}}{\sigma^2}$$

If the eigenvalue $\lambda_{B+1} \ll \sigma^2$ at the "cut-off point" $B+1$, we may expect the Nyström method to work correctly. Otherwise, for slowly decaying eigenvalues and/or small σ^2, the weight vector $\mathbf{w}$ may be far off its correct values.

A closer look at the Nyström approximation [9] revealed that already for moderately complex data sets, such as KIN8NM, it tends to underestimate eigenvalues of the Gram matrix, unless a very high number of basis points is used. If in addition a rather low noise variance is assumed, we obtain a high value for the error bound in Eq. (5.2), confirming our observations in the experiments.

6 Conclusions

Our results indicate that, depending on the computational resources and the desired accuracy, one may select methods as follows: If the major concern is speed of prediction, one is well advised to use the subset of representers method with basis selection by greedy posterior approximation (SRM GP). This method may be expected to give results that are significantly better than other (inductive) methods. While being painfully slow during basis selection, the resulting models are compact, easy to use and accurate.

On the other hand, if accurate predictions are the major concern, one may expect best results with the Bayesian committee machine. On large low noise data sets (such as KIN40K and ART) we observed significant advantages in terms of prediction accuracy, giving an average mean squared error that was only a fraction (25-30%) of the error achieved by the best inductive method. For the BCM, one must take into account that it is a transduction scheme, thus prediction time and memory consumption are larger than those of SRM methods.

We observed that reduced rank approximation with Nyström is not recommendable in its current form. We have provided some theoretical insight into problems with Nyström, methods to overcome these problems are currently investigated [9].

Although all discussed approaches scale linearly in the number of training data, they exhibit significantly different runtime in practice. For the experiments we had done in this paper (running 10-fold cross validation on given data) the Bayesian committee machine is on the order of one magnitude slower than an SRM method with random basis, SRM with greedy posterior approximation is again an order of magnitude slower than the BCM.

References

[1] Blake, C. and Merz, C. UCI repository of machine learning databases. 1998.

[2] Horn, R. A. and Johnson, C. R. *Matrix Analysis*. Cambridge University Press, 1985.

[3] Leen, T. K., Dietterich, T. G., and Tresp, V., eds. *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[4] MacKay, D. J. Introduction to gaussian processes. In C. M. Bishop, ed., *Neural Networks and Machine Learning*, vol. 168 of *NATO Asi Series. Series F, Computer and Systems Sciences*. Springer Verlag, 1998.

[5] Rasmussen, C. E. Reduced rank gaussian process learning, 2002. Unpublished Manuscript.

[6] Smola, A. and Sch¨olkopf, B. Sparse greedy matrix approximation for machine learning. In P. Langely, ed., *Proceedings of ICML00*. Morgan Kaufmann, 2000.

[7] Smola, A. J. and Bartlett, P. Sparse greedy gaussian process regression. In [3], pp. 619–625.

[8] Tresp, V. A bayesian committee machine. *Neural Computation*, 12:2719–2741, 2000.

[9] Williams, C. K., Rasmussen, C. E., Schwaighofer, A., and Tresp, V. Observations on the Nystr¨om method for Gaussian process prediction, 2002. Unpublished Manuscript.

[10] Williams, C. K. I. and Seeger, M. Using the nystr¨om method to speed up kernel machines. In [3], pp. 682–688.