Scalable Kernel Systems

Volker Tresp[1] and Anton Schwaighofer[1,2]

[1] Siemens AG, Corporate Technology, Otto-Hahn-Ring 6,
81739 München, Germany
{Volker.Tresp,Anton.Schwaighofer.external}@mchp.siemens.de
[2] TU Graz, Institute for Theoretical Computer Science
Inffeldgasse 16b, 8010 Graz, Austria
aschwaig@igi.tu-graz.ac.at
http://www.igi.tu-graz.ac.at/aschwaig/

Abstract. Kernel-based systems are currently very popular approaches to supervised learning. Unfortunately, the computational load for training kernel-based systems increases drastically with the number of training data points. Recently, a number of approximate methods for scaling kernel-based systems to large data sets have been introduced. In this paper we investigate the relationship between three of those approaches and compare their performances experimentally.

1 Introduction

Kernel-based systems such as the support vector machine (SVM) and Gaussian processes (GP) are powerful and currently very popular approaches to supervised learning. Kernel-based systems have demonstrated very competitive performance on several applications and data sets and have great potential for KDD-applications since their degrees of freedom grow with training data size and they are therefore capable of modeling an increasing amount of detail when appropriately many training data points become available. Unfortunately, there are at least three problems when one tries to scale up these systems to large data sets. First, training time increases drastically with the number of training data points, second, memory requirements increase with data set size and third, prediction time is proportional to the number of kernels and the latter is equal to (or at least increases with) the number of training data points. In this presentation, we will concentrate on Gaussian processes which are the basis for Gaussian process regression, generalized Gaussian process regression, and the support vector machine. We analyze and experimentally compare three recently introduced approaches towards scaling Gaussian processes to large data sets using finite-dimensional representations, thus obtaining learning rules which scale linearly in the number of training data points.

The first approach is the *subset of representers method* (SRM) and can be found in the work of Wahba [5], in the work on sparse greedy Gaussian process regression by Smola and Bartlett [2], and in the reduced support vector machine by Lee and Mangasarian [1]. The SRM is based on a factorization of the kernel functions. The second variant is a reduced rank approximation (RRA) of the Gram matrix introduced in the work of Williams and Seeger [6]. The RRA uses the same decomposition as the SRM but this decomposition is only applied to the Gram matrix. The third variant is the BCM approximation introduced by Tresp [3]. Here the starting point is the optimal projection of the data on a set of base kernels which requires the inversion of a covariance matrix of the size of the number of training data points. The BCM approximation is achieved by a block diagonal approximation of this matrix.

In this paper, we analyze the approaches from a common view point and we will compare the performances of the approximations where we pay particular attention to the issue of the optimal scale parameter.

The paper is organized as follows. In the next section, we will provide a brief introduction into Gaussian processes. In the following sections, we describe the SRM, the RRA and the BCM approximation. Section 6 analyzes the approximations and provides experimental comparisons. Section 7 contains the conclusions.

2 Gaussian Process Regression (GPR)

In GPR one assumes that *a priori* a function $f(x)$ is generated from an infinite-dimensional Gaussian distribution with zero mean and covariance $K(x_i, x_j) = cov(f(x_i), f(x_j))$ defined at input points x_i and x_j. Furthermore, we assume a set of training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where targets are generated according to

$$y_i = f(x_i) + \epsilon_i$$

where ϵ_i is independent additive Gaussian noise with variance σ^2. The optimal regression function $\hat{f}(x)$ takes on the form of a weighted combination of kernel functions

$$\hat{f}(x) = \sum_{i=1}^{N} w_i K(x, x_i). \tag{1}$$

Based on our assumptions, the maximum a posterior (MAP) solution for $w = (w_1, \ldots, w_N)'$ minimizes the cost function

$$\frac{1}{2} w' \Sigma w + \frac{1}{2\sigma^2} (\Sigma w - y)'(\Sigma w - y) \tag{2}$$

where $(\Sigma)_{i,j} = K(x_i, x_j)$ is the $N \times N$ Gram matrix. The optimal weight vector is the solution to a system of linear equation which in matrix form becomes

$$(\Sigma + \sigma^2 I)w = y. \tag{3}$$

Here $y = (y_1, \ldots, y_N)'$ is the vector of targets and I is the $N \times N$-dimensional unit matrix. The experimenter has to specify the positive definite kernel function. A common choice is that

$$K(x_i, x_j) = A \exp(-1/(2\gamma^2)||x_i - x_j||^2) \tag{4}$$

which is a Gaussian with positive amplitude A and scale parameter γ. Other positive definite covariance functions are also used.

3 The Subset of Representers Method (SRM)

Here, one first selects as set of N_b base kernels. These base kernels are typically either defined at a subset of the training data or of the test data. One can now approximate the covariance at x_i and x_j as

$$cov(f(x_i), f(x_j)) \approx (K^b(x_i))'(\Sigma^{b,b})^{-1} K^b(x_j). \tag{5}$$

Here, $\Sigma^{b,b}$ is the covariance matrix for the base kernel points and $K^b(x_i)$ is the vector of covariances between the functional values at x_i and the base kernel points. Since $\Sigma^{b,b}$ contains the covariances at the base kernel points, this approximation is an equality if either x_i or x_j are elements of the base kernel points and is an approximation otherwise. Note that using this approximation, the Gram matrix becomes

$$\Sigma \approx \Sigma^{m,b}(\Sigma^{b,b})^{-1}(\Sigma^{m,b})', \tag{6}$$

where $\Sigma^{m,b}$ contains the covariance terms between all N training data points and the base kernel points. With this approximation, the rank of the Gram matrix Σ cannot be larger than N_b. The regression function is now a superposition

$$\hat{f}(x) = \sum_{i=1}^{N_b} w_i K(x, x_i) \tag{7}$$

of only N_b kernel functions and the optimal weight vector minimizes the cost function

$$\frac{1}{2}(w^b)'\Sigma^{b,b}w^b + \frac{1}{2\sigma^2}(\Sigma^{m,b}w^b - y)'(\Sigma^{m,b}w^b - y), \tag{8}$$

where $w^b = (w_1, \ldots, w_{N_b})'$, and where y is the vector of all training targets. Note that the number of kernels is now N_b instead of N, hence the name subset of representers method (SRM). [1]

Usually, the base kernels are selected from the training data set either randomly or using a clustering algorithm [5]. Smola and Bartlett [2] select an (nearly) optimal subset of base kernel points out of the training data set. Their base kernel point selection procedure does not significantly increase the computationally complexity of the training procedure.

4 A Reduced Rank Approximation (RRA)

In a paper by Williams and Seeger [6], the authors use the decomposition of the Gram matrix of Equation 6 for calculating the kernel weights (Equation 3). Using standard matrix algebra (Woodbury formula), one obtains

$$w_{opt} \approx \frac{1}{\sigma^2}\left(y - \Sigma^{m,b}\left[(\Sigma^{m,b})'\Sigma^{m,b} + \sigma^2\Sigma^{b,b}\right]^{-1}(\Sigma^{m,b})'y\right).$$

[1] Incidentally, the relationship between the full kernel weights and the reduced kernel weights is given by $w^b = (\Sigma^{b,b})^{-1}(\Sigma^{m,b})'w$. Substitution of this identity in the cost function of Equation 8 and using Equation 5 leads to the cost function of Equation 2.

In the SRM method, the decomposition of Equation 5 changes the covariance structures of the kernels, whereas here, the covariance structures defining the kernels are unchanged. The factorization of the Gram matrix is used to obtain an efficient approximation of the optimal kernel weights. As a result, in the RRA approximation the number of kernels with nonzero weights is identical to the number of training data points N (Equation 1), whereas in the SRM method, the number of kernels with nonzero weights is identical to the number of base points N_b (Equation 7).

5 BCM Approximation

The Bayesian committee machine (BCM) was introduced by Tresp [3] and was derived using assumptions about conditional independencies. Here, we will choose a new approach to derive the BCM approximation. Let

$$P(f^b) = G(f^b; 0, \Sigma^{b,b})$$

be the Gaussian prior distribution of the unknown functional values at the base kernel points. Furthermore, let

$$P(y|f^b) = G(y; \Sigma^{m,b} w^b, cov(y|f^b))$$

be the conditional density of the training targets given f^b. Here, w^b is the weights vector defined on the base kernels, and

$$cov(y|f^b) = \sigma^2 I + \Sigma - \Sigma^{m,b}(\Sigma^{b,b})^{-1}(\Sigma^{m,b})' \tag{9}$$

is the covariance of the training data given f^b.

Note that both equations define a joint probability model and allow the calculation of many quantities of interest, e.g. $E(f^b|y)$. To be able to compare the BCM and the SRM, we will use the identity

$$f^b = \Sigma^{b,b} w^b. \tag{10}$$

The optimal w^b then minimizes the cost function

$$\frac{1}{2}(w^b)' \Sigma^{b,b} w^b + \frac{1}{2}(\Sigma^{m,b} w^b - y)' \; cov(y|f^b)^{-1} \; (\Sigma^{m,b} w^b - y). \tag{11}$$

Note that the errors in the likelihood term are correlated.

Equations 10 and 11 can be used to calculate the optimal prediction at the base kernel points but this requires the calculation of the inverse of $cov(y|f^b)$ and the latter has the dimension of $N \times N$. The BCM uses a block diagonal approximation of $cov(y|f^q)$ and the calculation of the weight vector w^b requires the inversion of matrices of only the block size B. The BCM approximation improves if few blocks are used (then a smaller number of elements are set zero) and when N_b is large, since then the last two terms on the right side of Equation (9) tend to cancel and $cov(y|f^b) \approx \sigma^2 I$. Note that the BCM approximation becomes the SRM if we set $cov(y|f^b) = \sigma^2 I$. In the latter, the induced correlations are completely ignored.

With the BCM approximation we obtain

$$w^b_{opt} \approx \left(\Sigma^{b,b} + \sum_{i=1}^{M} (\Sigma_i^{m,b})' cov(y^i|f^b)^{-1} \Sigma_i^{m,b} \right)^{-1} \sum_{i=1}^{M} (\Sigma_i^{m,b})' cov(y^i|f^b)^{-1} y^i$$

which is one particular form of the BCM approximation. Here, M is the number of blocks, y^i is the vector of targets of the i-th module, $cov(y^i|f^b)$ is the i-th diagonal block of $cov(y|f^b)$ and $\Sigma_i^{m,b}$ is the submatrix of $\Sigma^{m,b}$ containing the covariances between the base kernel points and the training data points in the i-th partition. The predictions at the base kernel points can be obtained by substituting w^b_{opt} in Equation 10. The predictions at additional test points can be calculated by substituting w^b_{opt} in Equation 7.

6 Experimental Comparisons

In the first experiment, the base points were selected out of the *test set*. This corresponds to a procedure sometimes referred to as transduction where the user starts training the system only after the inputs to the test points become available. The experimental results presented in Figure 1 (A), (B) show that for the optimal scale parameters, the BCM approximation makes significantly better predictions at the base kernel points if compared to the SRM. For large scale parameters, $cov(y|f^q)$ is approximately diagonal and both approximations give comparable results. For smaller scale parameters, the block diagonal approximation is better than the diagonal approximation and the BCM gives better results than the SRM. The predictions of the RRA at the base points are identical to the predictions of the SRM method. Based on the predictions at the base points, one can calculate prediction at additional test point. The results shown in Figure 1 (C) show that the results of the BCM and the SRM method are comparable, although the latter gives slightly better results.[2] Figure 1 (D) shows the test set error if the standard procedure is used. Here, the base points are randomly chosen out of the *training data set*, weights on the base points in the training data set are calculated, and these are then used to predict at the test points. As expected, the results shown in (D) are comparable to the results in (C). For the experiments in (C) and (D), the results of the RRA (not shown) were considerably worse than the results obtained using the BCM approximation and the SRM method.

[2] Of course one could calculate the BCM approximation again for the additional test points instead of using Equation 1. This would give better results but would require another $\mathcal{O}(Nm^2)$ operations.

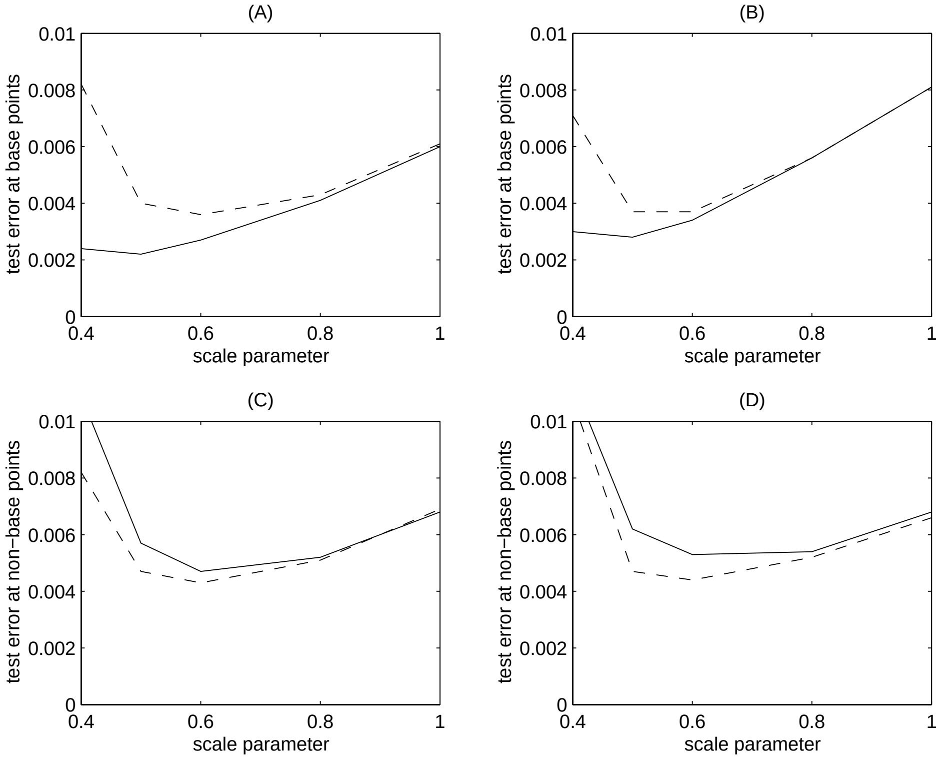

Fig. 1. Test set error is plotted against the scale parameter (width γ) of a Gaussian kernel for the BCM (continuous) and for the SRM (dashed). For (A), (B), and (C), the base points were randomly selected out of the test set. (A) and (B) show the performance at the base points and (C) shows the performance at additional test points. For the experiment in (D), the base points were randomly selected out of the training data set and the error on an independent test set is shown. We used 10000 training data points, 1000 base kernel points and 1000 additional test points. The plots are based on an artificial data set with additive noise with variance $\sigma^2 = 0$ (A, C, D) and $\sigma^2 = 0.001$ (B). The test data are noise free.

7 Conclusions

In this paper, we have compared three approaches for scaling up kernel-based systems. The computational complexity of the presented methods scales as $\mathcal{O}(N \times N_b^2)$ where N is the number of training data points and N_b is the number of base kernel points. If training is performed after the test inputs are known (transduction), the BCM outperforms the other approaches. In the more common setting where training is done before the inputs to the test set are available (induction), all three methods perform comparably, although the subset of representers method seems to have a slight advantage in performance.

References

1. Lee, Y.-J. and Mangasarian, O. L.: RSVM: Reduced Support Vector Machines. Data Mining Institute Technical Report 00-07, Computer Sciences Department, University of Wisconsin (2000)
2. Smola, A. J. and Bartlett, P.: Sparse Greedy Gaussian Process Regression. In: T. K. Leen, T. G. Diettrich and V. Tresp, (eds.): Advances in Neural Information Processing Systems 13 (2001)
3. Tresp, V.: The Bayesian Committee Machine. Neural Computation, Vol.12 (2000)
4. Tresp, V.: Scaling Kernel-Based Systems to Large Data Sets. Data Mining and Knowledge Discovery, accepted for publication
5. Wahba, G.: Spline models for observational data. Philadelphia: Society for Industrial and Applied Mathematics (1990)
6. Williams, C. K. I. and Seeger, M.: Using the Nyström Method to Speed up Kernel Machines. In: T. K. Leen, T. G. Diettrich and V. Tresp, (eds.): Advances in Neural Information Processing Systems 13 (2001)