Gaussian Process Regression: Active Data Selection and Test Point Rejection

Sambu Seo Marko Wallat Thore Graepel Klaus Obermayer
Department of Computer Science, Technical University of Berlin
Franklinstr.28, FR2-1, 10587 Berlin, Germany {*sontag,mawa,graepel2,oby*}*@cs.tu.berlin.de*

Abstract

We consider active data selection and test point rejection strategies for Gaussian process regression based on the variance of the posterior over target values. Gaussian process regression is viewed as transductive regression that provides target distributions for given points rather than selecting an explicit regression function. Since not only the posterior mean but also the posterior variance are easily calculated we use this additional information to two ends: Active data selection is performed by either querying at points of high estimated posterior variance or at points that minimize the estimated posterior variance averaged over the input distribution of interest or — in a transductive manner — averaged over the test set. Test point rejection is performed using the estimated posterior variance as a confidence measure. We find for both a two-dimensional toy problem and for a real-world benchmark problem that the variance is a reasonable criterion for both active data selection and test point rejection.

1 Introduction

The problem of regression, i.e. function estimation from given data, receives a lot of attention not only in the statistics literature but also in the neural network and machine learning communities. In addition to the task of finding a good regressor for a given data set we may consider two other related questions: i) How can the training data be selected efficiently? ii) What kind of performance guarantees can be given? Question i) is important whenever training data are difficult or expensive to obtain as is the case in many industrial applications where data points may correspond to test runs of plants under certain parameter settings or to expensive drilling operations in mining. Question ii) is relevant when dealing with risk sensitive applications such as medical or financial analysis. Gaussian Process (GP) regression is a flexible method to deal with nonlinear regression problems. Although its history can be traced back to the geophysical method of "krieging" GPs have recently been introduced to the neural network community as a "replacement for supervised neural networks" [5], in particular, because they can be viewed as a particular limit case of them [6]. The problem of (possibly non-linear) regression can be stated as follows: Assume we are given some noisy data $D = \{(x_i, t_i)\}_{i=1}^N, x_i \in \mathcal{X} = \mathbb{R}^L, t_i \in \mathcal{T} = \mathbb{R}$, for all $i \in \{1, \ldots, N\}$, where N is number of data points and L is the dimensionality of input vectors. Let D be drawn iid from a probability density $p(x,t) = p(t|x)p(x)$. Find a regression function $f \in \mathcal{F}, f : \mathcal{X} \mapsto \mathcal{T}$ such that the risk $\mathbf{E}_{\mathcal{X}\mathcal{T}}[l(f(x),t)]$ is minimized, where $l : \mathcal{T} \times \mathcal{T} \mapsto \Re^+$ specifies the pointwise regression loss, in our case the quadratic loss $l(t_1, t_2) = (t_1 - t_2)^2$. Gaussian Process regression deviates subtly from the standard formulation above because it is really a transductive method [3, 7] that does not provide a single regression function $f \in \mathcal{F}$ but a posterior density over target values for the test or working set. A Gaussian process is a collection of random variables $\mathbf{t} = (t(x_1), t(x_2), \ldots)$ which have a Gaussian joint distribution,

$$P(\mathbf{t}|\mathbf{C}, \mathbf{x_n}) = \frac{1}{Z} \exp\left(-\frac{1}{2}(\mathbf{t} - \mu)^T \mathbf{C}^{-1} (\mathbf{t} - \mu)\right) \tag{1}$$

for any set of inputs $\{\mathbf{x}_n\}$. $\mathbf{C}$ is the covariance matrix defined by the covariance function $C(x_n, x_m; \Theta)$ parameterized by hyperparameters Θ, and μ is the mean function.

Gaussian process regression makes a prediction $\tilde{t}$ on the new data point $\tilde{x}$ giving predictive mean and variance of the posterior distribution (for a derivation see [5]).

$$\begin{aligned} \hat{y}(\tilde{x}) &= \mathbf{k}(\tilde{x})\mathbf{C}_N^{-1}\mathbf{t} & (2) \\ \sigma^2_{\hat{y}}(\tilde{x}) &= C(\tilde{x},\tilde{x}) - \mathbf{k}(\tilde{x})\mathbf{C}_N^{-1}\mathbf{k}(\tilde{x}), & (3) \end{aligned}$$

where $\mathbf{k}(\tilde{x}) = (C(x_1,\tilde{x})\ldots,C(x_N,\tilde{x}))$ is the covariance between the training data and $\tilde{x}$, and $\mathbf{C}_N$ is the $N \times N$ covariance matrix of training data points given the covariance function C. The vector $\mathbf{C}_N^{-1}\mathbf{t}$ is independent of the new data and can be understood as representing the model constructed from the covariance function and the training data. Thus the simple Gaussian Process prior over functions makes a fully Bayesian analysis possible that results in the calculation of the mean and variance of the posterior density over target values, a procedure that may be called Bayesian Transduction [3]. It turns out that the posterior variance can be used in two useful ways: As a criterion for active data selection and as a measure of confidence in the prediction that may serve to reject test data. The remainder of this paper is structured as follows: In Section 2 we will describe two ways of how active data selection can be performed in the Gaussian Process regression mode. In Section 3 we describe a simple strategy for the rejection of test points to improve the generalization error on the remaining points. In Section 4 we demonstrate the effectiveness of the presented strategies on a 2-d toy example and on benchmark data from the DELVE data repository.

2 Active Learning with Gaussian Processes

Regression learning is usually based on the assumption that the data is provided in advance and that the learner is only a passive recipient (see e.g. [7]). In contrast, in the active learning scenario the learner may choose to query data points based on previously seen training data so as to incorporate as much new information into the model as possible [1]. A somewhat weaker model of active learning is that of query filtering or query selection, where the learner may select its queries from a stream or set of data points, respectively. In any case a criterion for the utility of querying a given point is required.

Let us consider two criteria for active data selection which are both based on the assumption that the given model (i.e. in our case the covariance function and noise model) is correct. The first method suggested by McKay [4] aims at maximizing the expected information gain about the parameter values of the model by selecting the data where the predictor exhibits maximum variance. We will refer to this active learning method as ALM. The idea can be applied to Gaussian Processes in a straight forward manner because the variance estimate $\sigma^2_{\hat{y}}(\tilde{x})$ for a query candidate $\tilde{x}$ is easily obtained using (3). Thus, in the query selection scenario one can use this criterion to choose the most promising candidate from a set of available points. Alternatively, one can perform an optimization on $\sigma^2_{\hat{y}}(\tilde{x})$, e.g., gradient ascent.

The second method suggested by Cohn [1] is motivated from the goal of minimizing the generalization error and will be refered to as ALC. To this end let us decompose the mean-square error (MSE) into a variance and a bias term.

$$E_{\mathrm{MSE}} = \underbrace{\sigma^2_{\hat{y}}}_{\text{variance}} + \underbrace{\mathbf{E}_{\mathcal{X}}\left[\left(\mathbf{E}_{\mathcal{T}}\left[\hat{y}(x)\right] - y(x)\right)^2\right]}_{\text{bias}^2}. \qquad (4)$$

Assuming that the model is correct we expect the bias to be small compared to the variance contribution. Thus in order to minimize the MSE we should aim at choosing our query $\tilde{x}$ such that the overall variance of the estimator is minimized.

For Gaussian Processes the variance of the output at a single point is given by (3). The effect of a query candidate $\tilde{x}$ on the overall variance can be estimated using the resulting $(N+1) \times (N+1)$ covariance matrix $\mathbf{C}_{N+1}$

$$\mathbf{C}_{N+1} = \begin{bmatrix} \mathbf{C}_N & \mathbf{m} \\ \mathbf{m}^{\mathbf{T}} & C(\tilde{x},\tilde{x}) \end{bmatrix} \quad \mathbf{C}_{N+1}^{-1} = \begin{bmatrix} \begin{bmatrix} \mathbf{C}_N^{-1} + \frac{1}{\mu}\mathbf{g}\mathbf{g}^T \end{bmatrix} & \mathbf{g} \\ \mathbf{g}^{\mathbf{T}} & \mu \end{bmatrix} \qquad (5)$$

where $\mathbf{m} = [C(x_1,\tilde{x})\ldots C(x_N,\tilde{x})] \in \mathbb{R}^N$ is the N-vector of covariances between the present training data points and the query candidate $\tilde{x}$. We used the partitioned inverse equations for the inverse of $\mathbf{C}_{N+1}$ [5],

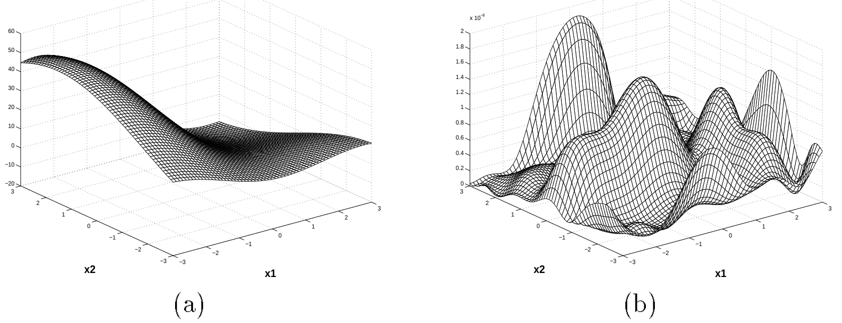


Figure 1: Illustration of Cohn's criterion (ALC) for active data selection. (a) Target function drawn from Gaussian prior with covariance function $C(x_n, x_m) = \exp\left(-\frac{1}{2}\sum_{i=1}^{2}\frac{(x_n^i - x_m^i)^2}{r_i^2}\right) + 0.01\, x_n x_m$, $r_1 = 3$, $r_2 = 2$. (b) Expected change of average variance as a function of query candidates $\tilde{x}$ averaged over 100 random reference data points ξ. 100 training data points and additive Gaussian noise ($\sigma^2 = 1$).

and

$$\mathbf{g} = -\mu \mathbf{C_N^{-1} m}, \quad \mu = \left(C(\tilde{x}, \tilde{x}) - \mathbf{m}^T \mathbf{C}_N^{-1}\mathbf{m}\right)^{-1} .$$

Now we can calculate how the output variance $\sigma^2_{\hat{y}(\xi)}$ on a reference data point ξ changes as a function of the query candidate $\tilde{x}$ if added to the training set.

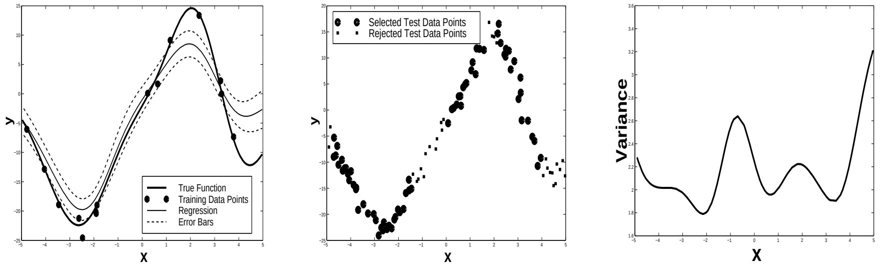
$$\triangle\sigma^2_{\hat{y}(\xi)}(\tilde{x}) = \sigma^2_{\hat{y}(\xi)} - \sigma^2_{\hat{y}(\xi)}(\tilde{x}) = \frac{\left(\mathbf{k}_N \mathbf{C}_N^{-1}\mathbf{m} - C(\tilde{x}, \xi)\right)^2}{\left(C(\tilde{x}, \tilde{x}) - \mathbf{m}^T \mathbf{C}_N^{-1}\mathbf{m}\right)}$$

where $\mathbf{k}_N = [C(x_1, \xi), \ldots, C(x_N, \xi)] \in \mathbb{R}^N$ is the vector of covariances between the training data and a reference data point ξ. Ideally, the change in variance $\triangle\sigma^2_{\hat{y}(\xi)}(\tilde{x})$ should be averaged over the input density $p(x)$ or over a density $q(x)$ that characterizes the importance of different regions in input space. In practice, $\triangle\sigma^2_{\hat{y}(\xi)}(\tilde{x})$ is averaged over an ensemble of data points sampled from $p(x)$ or $q(x)$, respectively. Of course, the average can be calculated w.r.t. the test set to be labeled in the spirit of transduction. Again, for the case of active data selection the gradient of $\triangle\sigma^2_{\hat{y}(\xi)}(\tilde{x})$ is easily calculated and can be used for optimization. For an illustration of the change of variance criterion, consider Figure 1. Note, that the averaged change of variance can be a multi-modal function of the query candidates, thus limiting the usefulness of greedy optimization algorithms.

With regard to both methods presented one should note that the target values that result from the queries do not enter the decision about which data point is queried next. This is a peculiarity of Gaussian Process regression and is a consequence of their linearity in the reproducing kernel Hilbert space spanned by the eigenfunctions of the covariance kernel. On the one hand, this feature makes active learning with Gaussian processes somewhat boring, because the optimum query strategy can be found in advance without seeing any of the target values of the training data. On the other hand, the query strategy still depends on both the covariance function and the input data already present and — in the case of ALC — on the "distribution of interest" $q(x)$, thus making the task sufficiently interesting.

3 Test Point Rejection for Gaussian Processes

According to Vapnik [7] transduction is a less ambitious task than induction because it aims at target values only at a finite number of given points instead of predictions on the whole of input space. It has been argued that this restriction may be able to improve the generalization ability of the method. However, another interesting feature of transduction is that a confidence measure can be given for individual predictions instead of predictors. How useful this kind of predictionwise confidence is, can be appreciated on as simple an example as weather forecasting. If you were to plan a hiking trip, would you be interested in a general



(a) (b) (c)

Figure 2: Illustration of test point rejection. Given the true function and the GP regression (see (a)) points from the test set are accepted or rejected (see (b)) according to the variance criterion (see(c)). The covariance function was given by $C(x_n, x_m) = \exp\left(-\frac{(x_n - x_m)^2}{2}\right)$. 15 data points were used for training and the rejection rate for the 100 test points was 0.3. Decrease in generalization error from 15.13 to 8.64.

evaluation of the quality of your weather station or would you rather know how good their actual prediction is in your case? The posterior variance given in (3) provides a confidence measure for estimated target values, given a correct model. This posterior variance is small if the covariance between the training data and the test data point is large. Intuitively, the covariance function acts as a similarity measure on the data space. High confidence, i.e. small posterior variance, is achieved when many of the training data are located near the test data point w.r.t. the covariance function. For an illustration of this effect consider Figure 2. Clearly, areas with few training examples exibit the highest variance and thus the lowest confidence.

Unfortunately, it is not clear, how exactly this confidence measure is related to the generalization error, especially, when the model is not exactly correct. However, we can use the variance in order to select, which test points we would like to reject at a given rejection rate. If the variance is predictive w.r.t. the generalization error, the error is expected to be a monotonically decreasing function of the rejection rate, a behaviour that is indeed observed and will be demonstrated in the experimental section.

4 Experimental Results

4.1 Active Data Selection

We performed numerical simulations in order to investigate inhowfar active data selection can be used in Gaussian Process regression. Furthermore we compared the two methods ALM and ALC based on the principles outlined by MacKay and Cohn, respectively. The generalization error as a function of the number of training data points serves as the measure of effectiveness and is calculated on held-out data. First, we tested the two schemes in a controlled experiment on a regression toy data set. Second, we applied the two methods to a real-world benchmark data set, were the exact form of the generating process was unknown.

The toy example we used is illustrated in Figure 1. The key feature of this experiment lies in the fact that we generate the data from a Gaussian process of known parameters and a prespecified noise model. The Covariance function was given by $C(x_n, x_m) = \exp\left(-\frac{1}{2}\sum_{i=1}^{2}\frac{(x_n^i - x_m^i)^2}{r_i^2}\right) + 0.01\, x_n x_m$, $r_1 = 3$, $r_2 = 2$ and we added Gaussian noise of variance $\sigma^2 = 1$. The first data point was chosen randomly while the subsequent 150 data points were actively selected using the ALM and the ALC methods. 500 reference data points were used for the evaluation of the average change in variance for ALC and were drawn from the uniform distribution over $[-3, +3] \times [-3, +3]$. The optimum query was found by evaluating the ALM and ALC criteria for 300 randomly drawn data points. Figure 3 shows the results averaged over 20 runs. Shown are both the variance and the MSE as a function of the number of data points — both quantities evaluated at 500 randomly drawn points. With regard to the MSE both ALC and ALM perform better than random selection, in particular in the beginning with few data points available. It should be noted, however, that ALC — as expected — decreases the variance more than both ALM and random selection do. As a more realistic example we used the data set `pumadyn-8nm` from the family of "pumadyn" data sets, which were generated synthetically

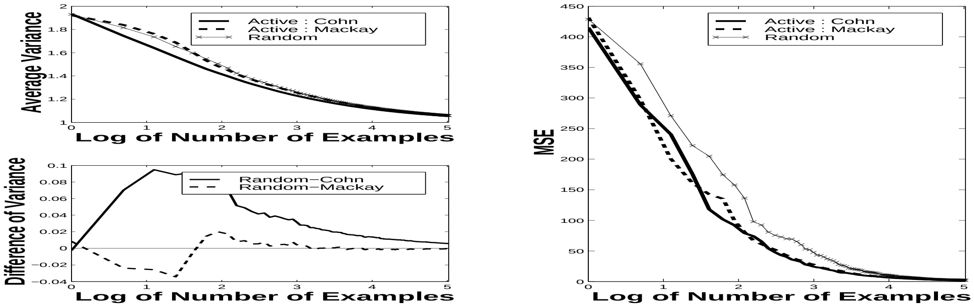


Figure 3: Learning curves for the 2d regression toy problem from Figure 1. Shown are the average variance (left) and MSE (right) for ALM, ALC and random selection. The plots show the average of 20 runs.

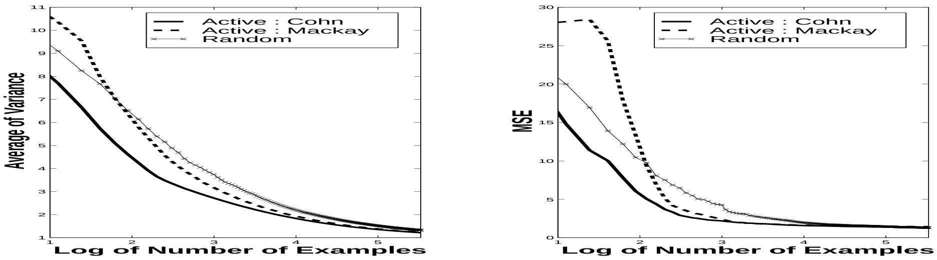


Figure 4: Learning curves for GP regression on `pumadyn-8nm`. Shown are the average variance (left) and MSE (right) for ALM, ALC and random selection. Plotted is the average over 26 runs for ALC and random selection and over 22 runs for ALM.

from a realistic simulation of the dynamics of a Puma 560 robot arm.[1] We chose a covariance function of the form $C(x^m, x^n) = \theta_2 + \theta_1 \exp\left[-\frac{1}{2}\sum_{i=1}^{D} w_i(x_i^m - x_i^n)^2\right] + \theta_4 \sum_{i=1}^{D} x_i^m x_i^n + \theta_3\delta(m,n)$, as suggested by [8] and determined suitable settings for the hyperparameters by the method of evidence maximization using cojugate gradient [8]. After a randomly drawn seed data point, 250 data points were queried according to the ALM and ALC criteria for 400 randomly drawn data points, and using random selection. Due to the size of the data set the criteria were applied only to a subset of the whole data set. The estimation of the generalization error was done on a hold-out set of size 500.

Figure 4 shows average variance and MSE as a function of the number of training data points. ALC performs consistently better than both ALM and random selection. As expected also the average variance is consistently lower for ALC. Interestingly, ALM performs worse than random selection for very few examples $N < 10$, at which point it becomes better than random selection. Apparently with very little information available the variance is not a good criterion for query selection. It should be noted, however, that the ALM criterion is much easier to evaluate than the ALC criterion and may thus be preferable in practical applications.

4.2 Test Point Rejection

On the same two data sets we explored if the variance is a suitable criterion for test point rejection in which case the generalization error is expected to be a monotonically decreasing function of the rejection rate.

From the toy problem we randomly generated 150 training data points and 1000 test data points for each of the 10 runs used for averaging. Figure 5 (left) shows the behaviour of the MSE as a function of rejection rate. The decrease in generalization error clearly indicates that the variance based rejection is a good strategy. The curve is steepest at low rejection rates and it is thus possible to achieve a good reduction in generalization error by rejecting only a small number of critical points.

[1] The data set and more information about it is available from: “http://www.cs.toronto.edu/delve/data/pumadyn/desc.html”

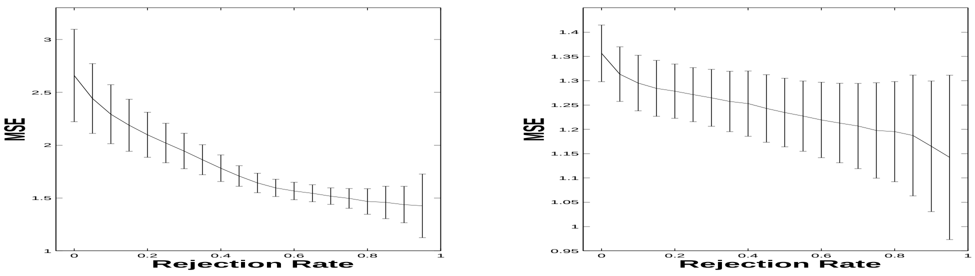


Figure 5: Generalization error as a function of rejection rate for the 2d toy data (left) and the `pumadyn-8nm` data (right). Shown are average and standard deviation over 50 runs. Parameters see text.

A similar experiment was performed on the `pumadyn-8nm` data set using the same parameter settings as for the active learning experiments. We used 250 data points for training and averaged over 50 runs. The remaining 7942 data points were held out for testing. Figure 5 (right) shows the MSE as a function of the rejection rate. The curve also shows a monotonic decrease in generalization error, however not as pronounced as in the toy example. This result can be explained by considering the model mismatch that is present in the real-world example and was avoided in the toy example, where the parameters of the underlying Gaussian process were known beforehand. Still, the variance appears to be a good predictor of generalization performance even in the real-world example.

4.3 Conclusion

Gaussian Process Regression — viewed as a transductive algorithm — lead us to criteria for active data selection and test point rejection. Cohn's criterion (ALC) of minimizing the average variance could be shown to perform well in the transductive mode where the variance is estimated and minimized at the test points. Mackay's criterion (ALM) for active data selection and the variance-based test point rejection were also demonstrated to accelerate and improve learning, respectively. These strategies exhibit a certain duality that can also be found in classification and is appears to be an interesting direction for future research to further explore and exploit this duality. Furthermore, from a learning theoretical point of view it would be desireable to be able to give bounds on the generalization error based on the variance, so as to establish a clear theoretical basis for the herein proposed methods.

References

[1] Cohn, D.A., *Neural networks exploration using optimal experiment design.* Neural Networks 6(9) ,1996.

[2] Gibbs, M.N., *Bayesian Gaussian processes for regression and classification.* PhD thesis, Cambridge University, 1997.

[3] Graepel,T., Herbrich R., and Obermayer, K., *Bayesian Transduction.* Advances in Neural Information Processing Systems 11, MIT Press, 1999.

[4] Mackay, D.J.C., *Information-based objective functions for active data selection*, Neural Computation, 4(4):589-603, 1992.

[5] Mackay, D.J.C., *Gaussian processes*, Tutorial at NIPS 10, 1998.

[6] Neal, R., *Monte Carlo implementation of Gaussian process Models for Bayesian regression and classification.* Technical Report No. 9702, Department of Statistics, University of Toronto, 1997.

[7] Vapnik, V., *Statistical Learning Theory.* New York, John Wiley and Sons, 1998.

[8] Rasmussen, C.E. and Williams, C.K.I. *Gaussian processes for regression* Advances in Neural Information Processing Systems 8, MIT press,1996.