Sensible Priors for Sparse Bayesian Learning

Joaquin Quiñonero-Candela Edward Snelson
Oliver Williams

Microsoft Research, Cambridge, UK
{joaquinc,esnelson,olliew}@microsoft.com

September 2007

Technical Report
MSR-TR-2007-121

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract

Sparse Bayesian learning suffers from impractical, overconfident predictions where the uncertainty tends to be maximal around the observations. We propose an alternative treatment that breaks the rigidity of the implied prior through decorrelation, and consequently gives reasonable and intuitive error bars. The attractive computational efficiency is retained; learning leads to sparse solutions. An interesting by-product is the ability to model non-stationarity and input-dependent noise.

1 Sparse Bayesian learning

Finite linear regression models are attractive for computational reasons and because they are easily interpreted. In these models, the regression function is simply a weighted linear sum of M basis functions $\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x})$:

$$f(\mathbf{x}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) \ , \tag{1}$$

where $\mathbf{x}$ is a (vectorial) input. A popular Bayesian treatment is the relevance vector machine (RVM) [1] in which a Gaussian prior is placed on the weights: $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, A)$, where A is a diagonal matrix of variance parameters $a_1, \ldots, a_M$. The observed outputs y are assumed to be corrupted by Gaussian white noise of variance σ^2 from the underlying regression function $f(\mathbf{x})$. Therefore, given a data set of N input-output pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, we can compute the Gaussian posterior distribution on the weights $p(\mathbf{w}|\mathbf{y})$ and make a Gaussian prediction at a new point $\mathbf{x}_*$: $p(y_*|\mathbf{x}_*, \mathbf{y})$. We give the prediction equations in appendix A. In the RVM model it is customary to use localized basis functions centered on the training inputs. The model evidence $p(\mathbf{y}|A)$ is maximized to learn the variances of the weights A. An attractive property of the RVM is that most of which tend to zero, effectively pruning the corresponding basis functions (for an explanation see [2]). The result is a sparse, and hence computationally efficient, linear model with $M \ll N$. The combination of a finite linear model with sparsity inducing priors on the weights is known as the *Sparse Bayesian learning* framework, and was inspired by the principle of automatic relevance determination (ARD) [3, 4].

As a Bayesian regression model, the RVM gives predictive distributions for new inputs, i.e., it supplies error bars, however these uncertainties are often unreasonable. This can be seen by examining the prior over functions given a set of basis functions: figure 1a shows a few sample functions drawn from the RVM prior given a small set of local (Gaussian) basis functions. Also shaded is the prior variance envelope (2 standard deviations shown). We see that the prior variance decays to zero away from basis function centres, and therefore the sample functions all return to the mean (zero). Hence using an RVM prior actually imposes strong constraints: the prior hypothesis space of functions

does not include any functions that vary away from the basis functions. Figure 1b shows a small set of data, and plots the mean RVM prediction along with the shaded predictive variance envelope having trained on this data. We see that the predictive variances drop away from the data, because there was no prior variance there. The behavior of the predictive variances is *opposite* to that desired — the RVM is most certain about the function far away from any observed data.

To better understand the limitations of the linear model above it is useful to consider the prior induced on the function values at N data points $\mathbf{f}$ directly. These have a zero mean Gaussian prior distribution, induced by the Gaussian prior on the weights, with covariance matrix:

$$K = \Phi_{NM} A \Phi_{NM}^\top \ , \tag{2}$$

where Φ_{NM} is the design matrix ($[\Phi_{NM}]_{nm} := \phi_m(\mathbf{x}_n)$). If there are more data points than basis functions this is a low-rank covariance matrix of rank M. In fact, the prior distribution on $f(\mathbf{x})$ is properly described as a Gaussian process (GP) (see e.g., [5]) with degenerate covariance function:

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top A \boldsymbol{\phi}(\mathbf{x}') \ . \tag{3}$$

Here, degenerate refers to the fact that any covariance matrix K formed from the covariance function $k(\mathbf{x}, \mathbf{x}')$ will have maximum rank M. The prior variance envelope as seen in figure 1a is given by the diagonal of the covariance function:

$$d(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) = \sum_{m=1}^{M} a_m \phi_m^2(\mathbf{x}) \ , \tag{4}$$

which decays to zero away from the support of the basis functions. A graphical representation of the covariance matrix K for the same basis functions as in figure 1a is shown in figure 2a.

Perhaps the most natural way to obtain sensible predictive variances is to use a full non-parametric Gaussian process model with a non-degenerate covariance function, such as the stationary Gaussian: $k(\mathbf{x}, \mathbf{x}') = \exp(-|\mathbf{x} - \mathbf{x}'|^2/\lambda^2)$, which has a constant diagonal. With this GP the prior variances are constant over all space, and consequently the predictive variances have the desired behavior — they grow large away from observed data. This can be seen as an infinite linear model which places basis functions everywhere in input space. However, by moving to a full GP the sparsity of the RVM model is lost, and hence its computational advantage. A full GP costs N^3 time for training due to inversion of the covariance matrix, and the prediction cost per test case is N for the mean and N^2 for the variance. This is a major problem for larger data sets.

In [6], the predictive variances of the RVM are corrected by adding an extra basis function at test time at the location of the test input. This provides the necessary extra predictive uncertainty, but unfortunately comes with an unacceptable extra prediction cost of NM per test case for both the mean and variance, making it impractical for large data sets. The RVM costs only NM^2

for training, and M and M^2 for the predictive mean and variance respectively. We seek to obtain a more appropriate prior from a finite linear model without incurring any additional computational cost.

2 Normalization

One way to achieve a constant prior variance is a normalization of the covariance function:

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = c\frac{k(\mathbf{x}, \mathbf{x}')}{\sqrt{k(\mathbf{x}, \mathbf{x})k(\mathbf{x}', \mathbf{x}')}} = c\frac{k(\mathbf{x}, \mathbf{x}')}{\sqrt{d(\mathbf{x})d(\mathbf{x}')}} \; . \tag{5}$$

This is equivalent to a conformal transformation of the kernel $k(\mathbf{x}, \mathbf{x}')$ by the factor $1/\sqrt{d(\mathbf{x})}$ [7] and it is easy to see that $\tilde{k}$ produces a constant *variance* of c (its diagonal). Applying this normalization to the RVM covariance (3) is equivalent to normalizing the basis functions:

$$\tilde{\phi}_m(\mathbf{x}) = \phi_m(\mathbf{x})/\sqrt{d(\mathbf{x})} \; . \tag{6}$$

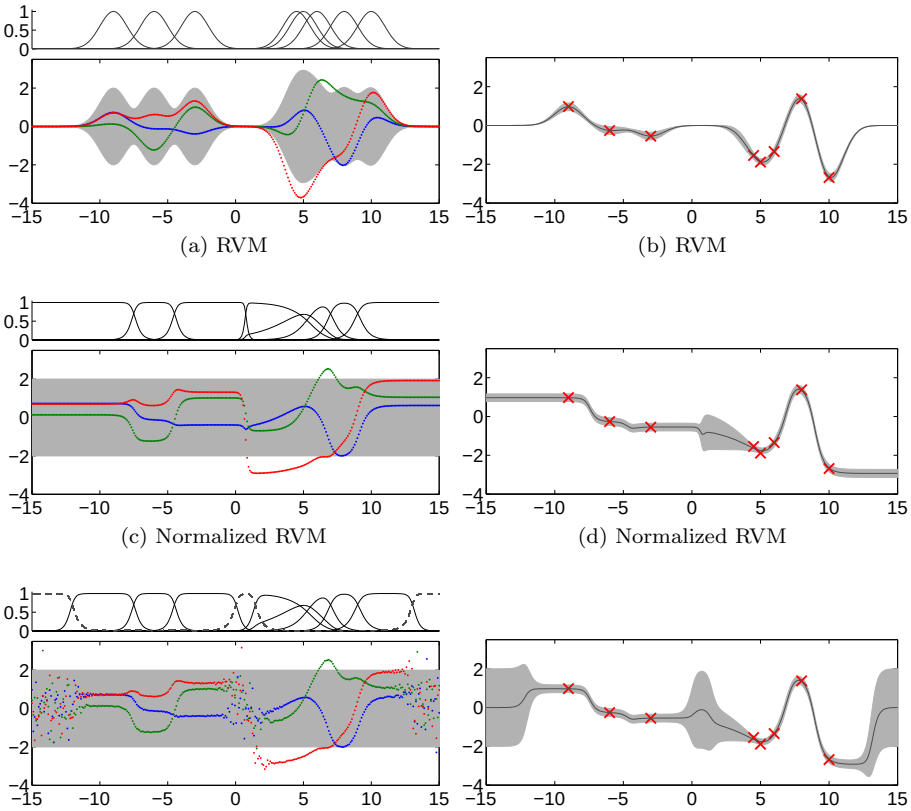
This is still therefore a finite linear model with M basis functions, however they have different shapes. This normalized model has a constant prior variance by construction, and so we might think that we have solved the problem of the RVM's unreasonable predictive variances.

Figure 1c shows the shapes of the normalized Gaussian basis functions along with some sample draws from this normalized prior. The function samples are no longer constrained to return to the mean away from basis function centers. However the normalization had flattened the basis functions, introducing long-term correlations away from their 'centers'. The predictive distribution in figure 1d shows a very different behavior to the original RVM, but still not what we wanted: the long-term correlations mean that the model is still extremely confident far away from observed data. Figure 2b shows how the long-term correlations appear as blocked regions of high covariance. Despite its constant diagonal, the covariance matrix is still low rank.

This highlights an important point: the problem with the RVM predictive variances is not due to the choice of local basis functions: however we change the shape of the basis functions we still have the same pathology, as was observed in [6]. The problem is fundamentally due to the finiteness of the model — the low-rank degenerate nature of the covariance — which does not give enough prior flexibility to functions; one can only draw at most M linearly independent functions from the corresponding GP prior.

3 Decorrelation and normalization

We would like to have a constant prior variance, but not at the expense of long-term correlations. We also want to preserve the computational sparsity of the RVM as compared to an infinite non-parametric GP model. Essentially we want the prior to decorrelate, but not decay, away from basis functions. To do



(a) RVM

(b) RVM

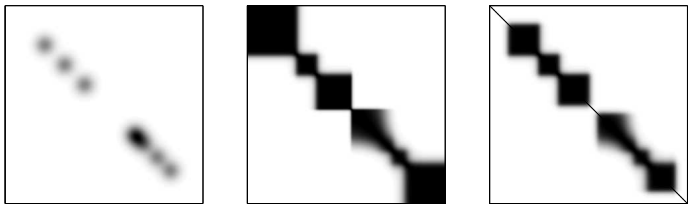
(c) Normalized RVM

(d) Normalized RVM

(e) Decorrelation and normalization. The envelope of the normalized white noise process is shown as the dashed line with the basis functions.

(f) Decorrelation and normalization

Figure 1: Left: samples from the prior, right: samples from the predictive distribution for a small training set drawn from the RVM prior, given by the crosses. Two standard deviation prior and predictive envelopes are shaded in gray. The basis functions, including normalization where appropriate, are shown above the prior plots. The parameters of the models are fixed to those of the generating RVM prior — no learning is taking place.



(a) RVM (b) Normalized RVM (c) Decor. & normaliz.

Figure 2: Covariance matrices. Dark areas indicate high covariance.

this we add a white noise Gaussian process $w_0(\mathbf{x})$ of constant variance a_0 to the linear model *before* normalization:

$$f(\mathbf{x}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}) + w_0(\mathbf{x}) \; . \tag{7}$$

The covariance function and its diagonal are now given by

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top A \boldsymbol{\phi}(\mathbf{x}') + a_0 \delta_{\mathbf{x},\mathbf{x}'} \quad \text{and} \quad d(\mathbf{x}) = \sum_{m=1}^{M} a_m \phi_m^2(\mathbf{x}) + a_0 \; . \tag{8}$$

Normalizing as above using (5) again gives a model with constant prior variance constituting a finite linear model with normalized basis functions plus a modulated white noise Gaussian process:

$$f(\mathbf{x}) = \sqrt{c}\Big[\sum_{m=1}^{M} w_m \frac{\phi_m(\mathbf{x})}{\sqrt{d(\mathbf{x})}} + w_0(\mathbf{x}) \frac{1}{\sqrt{d(\mathbf{x})}} \Big] \; . \tag{9}$$

The effect of this normalized white noise process can be seen in figure 1e, which shows samples from the prior for a particular choice of a_0. The basis functions are flattened to a degree, but unlike the noiseless normalized solution, as we move away from the basis function centers the white noise process takes over and decorrelates the sample functions. The original constant variance white noise process is normalized to have variance $a_0/d(\mathbf{x})$, which means it dominates when the basis functions decay. Its envelope is shown as a dashed line with the basis functions in figure 1e. The relative magnitude of a_0 to A determines the strength of the noise process relative to the basis functions. Figure 1f shows predictions from the model, where we now observe the desired behavior of the predictive variances which grow as we move away from data. Figure 2c illustrates the new covariance matrix, with constant diagonal and reduced blocks of high correlation.

By adding a weight *function* $w_0(\mathbf{x})$ to the model it might at first seem that this implies the addition of infinitely many new basis functions, and potentially

an increase in computational cost. However, since $w_0(\mathbf{x})$ is a white noise process, no additional correlations are introduced in the model, and hence the computational cost remains the same as for the RVM. A way to see this is to look at the covariance matrix:

$$\begin{aligned}\tilde{K} &= cD^{-\frac{1}{2}}KD^{-\frac{1}{2}} \\ &= c[\tilde{\Phi}_{NM}A\tilde{\Phi}_{NM}^{\top} + a_0 D^{-1}] \,,\end{aligned} \tag{10}$$

where $D = \mathrm{diag}[d(\mathbf{x}_1), \ldots, d(\mathbf{x}_N)]$ and $\tilde{\Phi}_{NM}$ are the normalized basis functions. This is no longer a low-rank covariance matrix, but rather a low-rank matrix plus a diagonal. Crucially the inversion of this matrix (plus the measurement noise $\sigma^2 I_N$) can still be performed in NM^2 time. Also the cost of the predictions remains the same as a finite linear model: M for the mean and M^2 for the variance per test case. Just like in the RVM we learn parameters of the model by maximizing the evidence $p(\mathbf{y})$ using gradient ascent. Details of the prediction equations and evidence are given in appendix A.

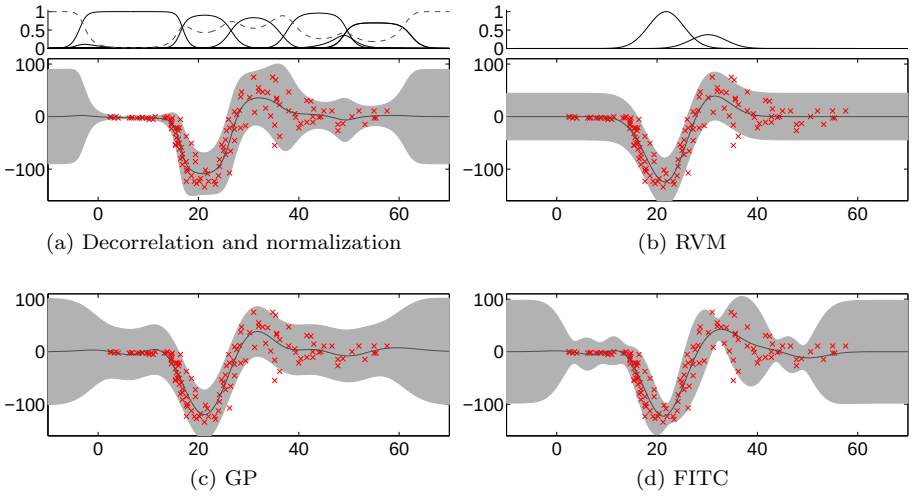
4 Sparse Gaussian processes

There is a strong relationship between our proposed solution to the RVM's predictive variance problems and sparse Gaussian process approximations. Sparse GP approximations start with a given covariance function, and seek to approximate this in order to reduce the N^3 complexity of the full GP. For the purposes of this paper will shall compare to the FITC approximation, as used by [8, 9] and reviewed in [10]. The FITC approximation is based on a small set of M support input points which determine the regions in which the approximation is good. FITC is most easily summarized by its prior covariance matrix:

$$K_{\mathrm{FITC}} = K_{NM}K_{MM}^{-1}K_{NM}^{\top} + \mathrm{diag}(K_{NN} - K_{NM}K_{MM}^{-1}K_{NM}^{\top}) \,. \tag{11}$$

Here K_{NM}, K_{MM}, and K_{NN} are covariance matrices constructed from the original covariance function to be approximated; N and M refer to the data points and the support points respectively.

Comparing (11) to (10) we note some obvious similarities and differences. They both consist of a low-rank matrix plus a diagonal, and hence if the number of basis functions is equivalent to the number of support points their computational costs are the same. Further similarities are apparent if we also choose the basis functions $\boldsymbol{\phi}(\mathbf{x})$ to be Gaussian and the underlying covariance function for FITC to be Gaussian i.e. $K_{NM} \equiv \Phi_{NM}$. Now both cases consist of a similar low-rank matrix which is 'corrected' to achieve a constant diagonal. In FITC this correction is additive purely on the diagonal. In our new approach the correction is divisive, and therefore alters the shapes of the basis functions too. Due to the constant diagonal, FITC shows a decorrelation effect in its prior as we move away from the FITC support points, similar to that shown in figure 1e.



(a) Decorrelation and normalization (b) RVM

(c) GP (d) FITC

Figure 3: Predictions for Silverman's motorcyle data set

There are a number of differences too. Our new construction allows us to choose an arbitrary set of basis functions, not necessarily derived from any kernel function, and normalize them to produce sensible priors. FITC requires an underlying desired GP covariance function for its construction. Secondly, just like in the original RVM, we can use the adjustable A variance parameters to automatically prune out unnecessary basis functions, thereby finding a very sparse solution. The number of support points in FITC must be set by hand.

A direct application of the FITC diagonal correction (11) to the linear model is not possible, as the covariance matrix would no longer be guaranteed to be positive definite.

5 A non-stationary heteroscedastic example

We tested our method on Silverman's motorcycle data set [11]; accelerometer readings as a function of time in a simulated impact experiment on motorcycle crash helmets, with 133 recordings. This is a classic benchmark dataset which exhibits both heteroscedastic (variable noise levels) and non-stationary properties. Figure 3a shows the result. We used Gaussian basis functions $\phi_m(\mathbf{x}) = \exp(-|\mathbf{x}-\mathbf{x}_m|^2/\lambda^2)$, and learnt the parameters of the model (A, a_0, c, λ, σ^2) by maximizing the evidence with gradient ascent as described in appendix A. Initially there was a basis function centred on every data point, but as the upper section of the plot shows, only a handful of significant basis functions remain after training: learning A prunes almost all of them away leaving a very sparse solution. Also note that the shapes of the remaining basis functions have changed through normalization, adapting well to the non-stationary aspects of

the data (for example the left-most flat section). Finally, the added noise process is modulated such that it not only gives uncertain predictions away from the data, but it also models very well the heteroscedastic noise in the data. Although our model was designed to fix the RVM's predictive variances, we find that the normalization also models non-stationarity and heteroscedasticity.

Figure 3b shows the trained RVM's predictions. Its noise level is constant, and so it cannot model the heteroscedasticity, and its predictive variances do not grow away from the data. A full GP with Gaussian covariance is shown in figure 3c. Again it can only learn a single gobal noise-level, and so it is not a good model for this data. Figure 3d shows the FITC sparse GP approximation, where we use 8 support points, which is also learnt as in [9]. This model is therefore of comparable sparsity to figure 3a. This is actually a better model of heteroscedasticiy than the full GP, because it has a similar input-dependent noise component as our new model. However it shows a tendency to overfit slightly by 'pinching in' at the support points, and its underlying Gaussian stationary covariance is too smooth to model the data well.

The motorycle data set has also been used to test other approaches to non-stationary and/or heteroscedastic GP regression [12, 13]. In contrast to the simple linear model discussed here, these approaches are computationally much more expensive, involving infinite mixtures and requiring sampling. Heteroscedastic GP regression is also addressed in [14], and other non-stationary GP covariance functions are discussed in [15], among others.

6 Real-time probabilistic visual tracking

An application in which both the sparsity of the RVM and meaningful probabilistic predictions are important is visual tracking. In [16] a *displacement expert* is created by training RVM regression to predict the true location of a target object given an initial estimate of its position in an image. This uses the pixel intensities sampled from the initial image region as (high dimensional) input vectors and as a consequence evaluating a basis function is expensive. By pruning many of the basis functions from the model, the RVM yields an extrememly efficient tracker.

The Gaussian RVM displacement predictions can be fused with a dynamical motion model over time with a Kalman filter, typically yielding improved accuracy. However, when a target changes appearance significantly or becomes occluded, the small variances predicted by the RVM corrupt the Kalman filter estimate of the state and consequently the tracker fails (see the top row of figure 4).

The bottom row of figure 4 shows the displacement expert tracking a target through an occlusion when the decorrelated and normalized linear model proposed in this paper is used. When the tracked person is occluded by a tree, the new model correctly makes predictions with a large variance which consequently contribute very little to Kalman filter state updates, which instead relies on the constant velocity dynamical model. Once the occlusion is over, the displace-

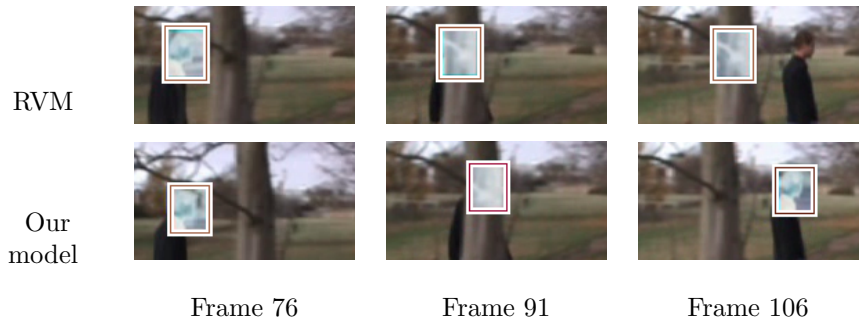


Figure 4: Examples of tracking a target undergoing occlusion. Top row: Implementing a displacement expert with a conventional RVM causes tracking to fail with the target is temporarily occluded by a tree. Bottom: Implementing the displacement expert with a decorrelated and normalized linear model provides meaningful predictive uncertainties which permit the Kalman filter to fall back on the dynamical prior during the occlusion.

ment expert is again able to make confident predictions and accurate tracking resumes.

The same successful tracking performance could be achieved by using a full GP, but this would come at a significantly higher computational cost, failing to meet real-time requirements. The difficulty with using FITC — a sparse GP approximation that produces sensible error bars — is that finding the inducing inputs requires an optimization in a space that is here of very high dimension.

7 Discussion

Sparse finite linear models are impractical from a probabilistic point of view, since independently of the type of basis functions used they tend to be overconfident, particularly for predictions away from the observations. Samples from the implied prior live in an at most M-dimensional function space; this severely restricts the available posterior uncertainty.

By incorporating an infinite set of uncorrelated basis functions to the model, we enrich the prior over functions. Normalization ensures a constant prior variance, and introduces decorrelations. The rôle of the initial localized basis functions is now to introduce local correlations, that do not overconstrain the posterior. The predictive variances increase away from the observed data, in a similar fashion as for non-parametric non-degenerate GPs.

The new model can still be treated as a finite linear model and retains the same propensity to sparsity as the RVM, with the corresponding computational advantage. This is due to the fact that the new basis functions do not correlate to anything, and the number of sources of correlation remains unchanged: M, the number of original basis functions. For large data sets, the computationally

efficient inference schemes that have been devised for the RVM [17] can be used.

The new treatment of finite linear models proposed makes them suitable for fitting non-stationary and heteroscedastic data. By individually varying the ratio of the M prior variances A to the variance a_0 of the uncorrelated process, the model can both change the shape of the basis functions and the level of input dependent noise.

8 Outlook

Although we have this far normalized assuming that the desired prior variance was constant, normalizing to achieve any arbitrary (and valid) prior envelope $c(\mathbf{x})$ is straightforward: the constant c is replaced by the function $c(\mathbf{x})$ (for instance in (5)). For example, if the prior variance of a model linear in the inputs was desired, $c(\mathbf{x})$ would be a quadratic form.

The way basis functions are chosen needs further investigation given that their shape is altered by normalization. For example, one could now consider using exponential basis functions, knowing that they will be bounded after normalization.

A Predictive distribution and evidence

All that is needed to make predictions with a finite linear model in general and with the RVM in particular, is the posterior over the M dimensional weights vector:

$$p(\mathbf{w}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma) \;, \quad \text{with} \quad \Sigma = (\Phi_{NM}^\top B^{-1} \Phi_{NM} + A^{-1})^{-1} \quad \text{and} \quad \boldsymbol{\mu} = \Sigma \Phi_{NM}^\top B^{-1} \mathbf{y} \;, \tag{12}$$

where $B = \sigma^2 I_N$ is a unit matrix of size N proportional to the variance of the measurement noise σ^2. Given a new test input $\mathbf{x}_*$ we first evaluate the response of all M basis functions Φ_{*M}, and use the posterior over the weights to obtain the mean and the variance of the Gaussian predictive distribution:

$$\mathrm{E}(f(\mathbf{x}_*)) = \Phi_{*M} \boldsymbol{\mu} \;, \quad \text{and} \quad \mathrm{Var}(f(\mathbf{x}_*)) = \Phi_{*M} \Sigma \Phi_{*M}^\top \;. \tag{13}$$

Although the normalized model we are proposing contains a weight process $w_0(x)$, to make predictions we only need to compute the posterior over the M weights associated to the original basis functions. The posterior is again Gaussian, with mean and covariance very similar to those of the RVM:

$$\tilde{\Sigma} = (\tilde{\Phi}_{NM}^\top \tilde{B}^{-1} \tilde{\Phi}_{NM} + c^{-1} A^{-1})^{-1} \quad \text{and} \quad \tilde{\boldsymbol{\mu}} = \tilde{\Sigma} \tilde{\Phi}_{NM}^\top \tilde{B}^{-1} \mathbf{y} \;, \tag{14}$$

but with a new definition of the diagonal noise variance matrix $\tilde{B} = \sigma^2 I_N + c a_0 D^{-1}$, and where the normalized basis functions are used $\tilde{\Phi}_{NM} = D^{-1/2} \Phi_{NM}$. We remind that $D = \mathrm{diag}(d(\mathbf{x}_1), \ldots, d(\mathbf{x}_N))$ with $d(\mathbf{x}) = a_0 + \sum_{m=1}^M a_m \phi_m^2(\mathbf{x})$.

In our proposed model, the mean and the variance of the predictive distribution are given by:

$$\mathrm{E}(f(\mathbf{x}_*)) = \tilde{\Phi}_{*M}\tilde{\boldsymbol{\mu}}\ , \quad \text{and} \quad \mathrm{Var}(f(\mathbf{x}_*)) = \tilde{\Phi}_{*M}\tilde{\Sigma}\tilde{\Phi}_{*M}^\top + \frac{ca_0}{d(\mathbf{x})}\ . \tag{15}$$

Although the expression for the predictive mean remains unchanged (up to normalization), the predictive variance gets an additional additive term that comes from the modulated white noise process.

For our model the evidence is an N-variate Gaussian distribution with zero mean, and covariance given by $\tilde{C} = \tilde{\Phi}_{NM}A\tilde{\Phi}_{NM}^\top + \tilde{B}$. Using the matrix inversion lemma, the negative log evidence can be written as:

$$\mathcal{L} = \frac{1}{2}\left[N\log(2\pi)+\log|cA|+\log|\tilde{B}|-\log|\tilde{\Sigma}|+\mathbf{y}^\top\tilde{B}^{-1}\mathbf{y}-\mathbf{y}^\top\tilde{B}^{-1}\tilde{\Phi}_{NM}\tilde{\Sigma}\tilde{\Phi}_{NM}^\top\tilde{B}^{-1}\mathbf{y}\right]. \tag{16}$$

The computational cost of evaluating the evidence is NM^2, as is that of computing its gradients with respect to the prior variances of the weights A, the prior variance a_0 of the w_0 process, the variance of the output noise σ^2, the prior overall variance of the function c, and the lengthscale λ of the isotropic Gaussian basis functions $\phi_m(\mathbf{x}) = \exp(-|\mathbf{x}-\mathbf{x}_m|^2/\lambda^2)$.

References

[1] M. E. Tipping. Sparse Bayesian learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[2] D. Wipf, J. Palmer, and B. Rao. Perspectives on sparse Bayesian learning. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, Massachussetts, 2004. The MIT Press.

[3] David J. C. MacKay. Bayesian non-linear modelling for the energy prediction competition. *ASHRAE Transactions*, 100(2):1053–1062, 1994.

[4] Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, Heidelberg, Germany, 1996.

[5] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT press, 2006.

[6] C. E. Rasmussen and J. Quiñonero-Candela. Healing the relevance vector machine by augmentation. In *International Conference on Machine Learning*, 2005.

[7] S. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[8] L. Csató and M. Opper. Sparse online Gaussian processes. *Neural Computation*, 14(3):641–669, 2002.

[9] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, Cambridge, Massachussetts, 2006. The MIT Press.

[10] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, Dec 2005.

[11] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *J. Roy. Stat. Soc. B*, 47(1):1–52, 1985.

[12] C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, 2002.

[13] E. Meeds and S. Osindero. An alternative infinite mixture of Gaussian process experts. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, Cambridge, Massachussetts, 2006. The MIT Press.

[14] P. W. Goldberg, C. K. I. Williams, and C. M. Bishop. Regression with input-dependent noise: A Gaussian process treatment. In Mi. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, Cambridge, Massachussetts, 1998. The MIT Press.

[15] C. Paciorek and M. Schervish. Nonstationary covariance functions for Gaussian process regression. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, Massachussetts, 2004. The MIT Press.

[16] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(8):1292–1304, 2005.

[17] M. E. Tipping and A. C. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In C. M. Bishop and B. J. Frey, editors, *Ninth International Workshop on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics, 2003.