Prediction at an Uncertain Input for Gaussian Processes and Relevance Vector Machines Application to Multiple-Step Ahead Time-Series Forecasting

Joaquin Quiñonero-Candela   Agathe Girard

Latest update November 13, 2002

1 Introduction

We assume a statistical model of the form

$$t = y + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

where t is the observable variable (target), y is the function output ($y = f(\mathbf{x})$) and the noise is an additive uncorrelated Gaussian white noise with variance σ^2.

Talk here about having uncertainty in the test inputs, make reference to approximate approach (agathenips02).

2 Gaussian Processes

2.1 The Gaussian Process prior

The Gaussian Process (GP) modeling framework consists in placing a Gaussian prior over the function space: $f(\mathbf{x}) \sim GP(m(\mathbf{x}), C_{\mathrm{GP}}(\mathbf{x}, \mathbf{x}'))$. That is, for each n, we assume $y_1, \ldots, y_n \sim \mathcal{N}(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma}_{pq} = C_{\mathrm{GP}}(\mathbf{x}_p, \mathbf{x}_q)$ gives the covariance between any pair of points.

Here, we assume that the process is stationary, i.e. it has a constant mean (we choose $m(\mathbf{x}) = 0$) and the covariance function depends only on the distance between the points. We define it in section 2.2.

2.2 Gaussian Kernel

A common choice of kernel or covariance function is the Gaussian one:

$$C_{\mathrm{GP}}(\mathbf{x}_p, \mathbf{x}_q) = \exp\left[-\frac{1}{2}(\mathbf{x}_p - \mathbf{x}_q)^T \mathbf{\Lambda}^{-1} (\mathbf{x}_p - \mathbf{x}_q)\right] , \tag{2}$$

where $\mathbf{x}$ is a $D \times 1$ vector.

In the GP case, we consider $\mathbf{\Lambda} = \boldsymbol{\lambda} I$, where I is the $D \times D$ identity matrix and $\boldsymbol{\lambda} = [\lambda_1^2, \dots, \lambda_D^2]^T$, allowing for a different distance measure in different input directions.

2.3 Prediction at $\mathbf{x}^*$

Given a set of training data $\mathcal{D} = [\mathbf{x}_i, t_i]_{i=1}^n$, the predictive distribution of y^* at a new input $\mathbf{x}^*$ is obtained by conditioning on the training data to obtain $p(y^*|\mathbf{x}^*, \mathcal{D})$.

In the Gaussian Process modeling framework, the joint probability distribution of y^* and the training data $\mathbf{t}$ is Gaussian with zero-mean and covariance matrix C_{GP} which we can write

$$C_{\mathrm{GP}} = \begin{bmatrix} \mathbf{K} & \mathbf{k}(\mathbf{x}^*) \\ \mathbf{k}(\mathbf{x}^*)^T & k \end{bmatrix} \tag{3}$$

where K is the $N \times N$ 'data covariance matrix', such that $\mathbf{K}_{pq} = \mathbf{\Sigma}_{pq} + \sigma^2 \delta_{pq}$, $k = C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}^*)$ and $\mathbf{k}$ is the vector of covariances between the new and the training inputs, $\mathbf{k}(\mathbf{x}^*) = [C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_1) \dots C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}$
By conditioning on the observed cases, we have

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{t} \tag{4}$$

$$\sigma^2(\mathbf{x}^*) = k - \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}^*) \tag{5}$$

where $\mu(\mathbf{x}^*)$ and $\sigma^2(x^*)$ are the mean and the variance of the Gaussian predictive distribution $p(y^*|\mathbf{x}^*, \mathcal{D})$.

3 The Relevance Vector Machine

The Relevance Vector Machine (RVM) is a probabilistic sparse kernel model, identical in functional form to the Support Vector Machine (SVM) model, which is

$$f(\mathbf{x}) = \sum_{j=1}^{M} \omega_j \phi_j(\mathbf{x}) + \omega_0 = \boldsymbol{\omega}^T \boldsymbol{\phi}(\mathbf{x}) + \omega_0 \tag{6}$$

where $\{\omega_j\}$ are the model weights and $\phi_j(.)$ is an arbitrary basis function. We also write in vector form the weights vector $\boldsymbol{\omega} = [\omega_1, \dots, \omega_M]^T$ and the responses of all basis functions $\boldsymbol{\phi}(\mathbf{x}) \equiv [\phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x})]^T$ to the input $\mathbf{x}$. In the RVM case, a prior is put over the weights, governed by a set of hyperparameters, one associated with each weight. For the specific choice of a factorized distribution with variance α_j^{-1}:

$$p(\omega_j|\alpha_j) = \sqrt{\frac{\alpha_j}{2\pi}} \exp\left(-\frac{1}{2}\alpha_j \, \omega_j^2\right) \tag{7}$$

the prior over functions $p(\mathbf{y}|\boldsymbol{\alpha})$ is $\mathcal{N}(0, \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T)$, i.e. a Gaussian process with covariance function given by

$$C_{\mathrm{RVM}}(\mathbf{x}_p, \mathbf{x}_q) = \sum_{j=1}^{M} \frac{1}{\alpha_j} \phi_j(x_p) \phi_j(x_q) \tag{8}$$

where $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_N)^T$ and $\mathbf{A} = \mathrm{diag}(\alpha_0, \ldots, \alpha_N)$, and matrix $\boldsymbol{\Phi}$ is such that $\boldsymbol{\Phi}_{pq} = \phi_q(\mathbf{x}_p)$. Sparseness in terms of the basis vectors may arise if for some j $\alpha_j^{-1} = 0$. Then the jth basis function will not contribute to the model. Associating a basis function with each input point may thus lead to a model with a sparse representation in the inputs, i.e. the solution is only spanned by a subset of all input points. This is exactly the idea behind the relevance vector machine.

3.1 Gaussian basis functions

One way of associating a basis function with each training input point is to choose (non-normalized) Gaussian basis functions of the form:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_j)^T \boldsymbol{\Lambda}^{-1}(\mathbf{x} - \mathbf{x}_j)\right) \tag{9}$$

where x_j are the training inputs, and the functions are isotropic with $\boldsymbol{\Lambda} = \lambda \mathbf{I}$.

The resulting covariance function is obtained by inserting expression (9) into equation (8), and is given by:

$$C_{\mathrm{RVM}}(\mathbf{x}_p, \mathbf{x}_q) = \sum_{j=1}^{M} \frac{1}{\alpha_j} \exp\left[-\frac{1}{2}\left((\mathbf{x}_p - \mathbf{x}_j)^T \boldsymbol{\Lambda}^{-1}(\mathbf{x}_p - \mathbf{x}_j) + (\mathbf{x}_q - \mathbf{x}_j)^T \boldsymbol{\Lambda}^{-1}(\mathbf{x}_q - \mathbf{x}_j)\right)\right] \tag{10}$$

One clear advantage of Gaussian basis functions is that they allow the exact analytical computation of the mean and variance of the predictive distribution for the case where the input is uncertain. These derivations are made in section 4.

Furthermore, it can be shown that for an infinite number of equally spaced Gaussian basis functions, equation (10) converges to the Gaussian covariance function of a GP, given by equation (2) [MacKay, 1997].

3.2 RVMs viewed as GPs

RVMs are Gaussian processes where the covariance between the training targets, based on equation (8), is given by the 'data covariance matrix' (see section 2.3) of the RVM:

$$\mathbf{K} = \sigma^2 \delta_{pq} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T \quad \text{or} \quad \mathbf{K}_{pq} = \sigma^2 \delta_{pq} + \sum_{j=1}^{M} \frac{1}{\alpha_j} \phi_j(\mathbf{x}_p)\phi_j(\mathbf{x}_q) \tag{11}$$

The vector of covariances between the new prediction and the training targets is given by

$$\mathbf{k}(\mathbf{x}^*) = \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\phi}^* \quad \text{or} \quad [\mathbf{k}(\mathbf{x}^*)]_p = \sum_{j=1}^{M} \frac{1}{\alpha_j} \phi_j(\mathbf{x}_p)\phi_j(\mathbf{x}^*) \tag{12}$$

where we set $\boldsymbol{\phi}^* = \boldsymbol{\phi}(\mathbf{x}^*)$. Finally, the covariance of the new prediction is given by $k = C_{\mathrm{RVM}}(\mathbf{x}^*, \mathbf{x}^*) = \boldsymbol{\phi}^{*T}\mathbf{A}^{-1}\boldsymbol{\phi}^*$.

Prediction of y^* at a new input $\mathbf{x}^*$ can be computed using the same approach as for GPs (section 2.3), by computing the joint distribution of y^* and the data first, and conditionning then on the data to obtain the predictive distribution $p(y^*|\mathbf{x}^*, \mathcal{D})$.

Plugging the expressions of $\mathbf{K}$, $\mathbf{k}(\mathbf{x}^*)$ and k for the RVM into equations (4) and (5) we obtain:

$$\mu(\mathbf{x}^*) = \boldsymbol{\phi}^{*T}\boldsymbol{\omega}_{MP} \tag{13}$$
$$\sigma^2(\mathbf{x}^*) = \boldsymbol{\phi}^{*T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\phi}^* \tag{14}$$

where $\boldsymbol{\omega}_{MP}$ and $\boldsymbol{\Sigma}$ are the mean and the variance of the posterior distribution over the weights. They are given by:

$$\boldsymbol{\omega}_{MP} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t} \tag{15}$$
$$\boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \mathbf{A})^{-1} \tag{16}$$

Equations (13)and (14) correspond to the classical expression of the mean and variance of the predictive distribution for the RVM [Tipping, 2001].

4 Prediction at $\mathbf{x}^* \sim \mathcal{N}(\mu_{\mathbf{x}^*}, \boldsymbol{\Sigma}_{\mathbf{x}^*})$

The predictive distribution of the function value, $p(f(\mathbf{x}^*))$ at the random variable $\mathbf{x}^*$, with $\mathbf{x}^* \sim \mathcal{N}(\mu_{\mathbf{x}^*}, \boldsymbol{\Sigma}_{\mathbf{x}^*})$, is obtained by integrating over the input distribution (omitting the conditioning on the training data)

$$p(y^*|\mu_{\mathbf{x}^*}, \boldsymbol{\Sigma}_{\mathbf{x}^*}) = \int p(y^*|\mathbf{x}^*)p(\mathbf{x}^*)d\mathbf{x}^* , \tag{17}$$

where $p(y^*|\mathbf{x}^*) = \frac{1}{\sigma(\mathbf{x}^*)\sqrt{2\pi}} \exp\left[-\frac{1}{2}\frac{(y^*-\mu(\mathbf{x}^*))^2}{\sigma^2(\mathbf{x}^*)}\right]$ with mean and variance depending on the model.

4.1 Numerical approximation

Given that the integral (17) is analytically intractable ($p(y^*|\mathbf{x}^*)$ is a complicated function of $\mathbf{x}^*$), one possibility is to perform a numerical approximation of the integral by a simple Monte-Carlo approach:

$$p(y^*|\mu_{\mathbf{x}^*}, \boldsymbol{\Sigma}_{\mathbf{x}^*}) = \int p(y^*|\mathbf{x}^*)p(\mathbf{x}^*)d\mathbf{x}^* \simeq \frac{1}{T}\sum_{t=1}^{T} p(y^*|\mathbf{x}^{*t}) , \tag{18}$$

where $\mathbf{x}^{*t}$ are (independent) samples from $p(\mathbf{x}^*)$.

4.2 Gaussian approximation

The analytical Gaussian approximation consists in only computing the mean and variance of $y^*|\mu_{\mathbf{x}^*}, \boldsymbol{\Sigma}_{\mathbf{x}^*}$. They are obtained using respectively the law of iterated expectations and law of conditional variances:

$$
\begin{aligned}
m(\mu_{x^*}, \mathbf{\Sigma}_{x^*}) &= E_{\mathbf{x}^*}[E_{y^*}[y^*|\mathbf{x}^*]] = E_{\mathbf{x}^*}[\mu(\mathbf{x}^*)] &(19)\\
v(\mu_{\mathbf{x}^*}, \mathbf{\Sigma}_{x^*}) &= E_{\mathbf{x}^*}[\mathrm{var}_{y^*}(y^*|\mathbf{x}^*)] + \mathrm{var}_{\mathbf{x}^*}(E_{y^*}[y^*|\mathbf{x}^*]) \\
&= E_{\mathbf{x}^*}[\sigma^2(\mathbf{x}^*)] + \mathrm{var}_{\mathbf{x}^*}(\mu(\mathbf{x}^*)) &(20)
\end{aligned}
$$

where $E_{\mathbf{x}^*}$ indicates the expectation under $\mathbf{x}^*$.

4.2.1 Approximate solution

The approximate solution consists in approximating the mean and the variance of the predictive distribution by their Taylor expansion, of order 1 and 2 respectively. The details can be found in [Girard et al., 2003] and more extended in [Girard et al., 2002].

4.2.2 Exact solution

For deriving the following results, we use the fact that the mean and the variance of the predictive distribution for a deterministic input is given by very similar expressions both for GPs and RVMs. For the GP case we have:

$$\mu(\mathbf{x}^*) = \sum_i \beta_i C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_i) \tag{21}$$

$$\sigma^2(\mathbf{x}^*) = C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}^*) - \sum_i \sum_j \mathbf{K}_{ij}{}^{-1} C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_i) C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_j) \tag{22}$$

where we define $\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{t}$, and for the RVM case we have:

$$\mu(\mathbf{x}^*) = \sum_i \beta_i \phi_i(\mathbf{x}^*) \tag{23}$$

$$\sigma^2(\mathbf{x}^*) = \sum_i \sum_j \mathbf{\Sigma}_{ij}{}^{-1} \phi_i(\mathbf{x}^*) \phi_j(\mathbf{x}^*) \tag{24}$$

with $\boldsymbol{\beta} = \boldsymbol{\omega}_{MP}$, as given by (15). It is worth noticing that $C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_i)$ and $\phi_i(\mathbf{x}^*)$ are given by the same expression:

$$C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_i) = \phi_i(\mathbf{x}^*) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_j)^T \mathbf{\Lambda}^{-1} (\mathbf{x} - \mathbf{x}_j)\right) \tag{25}$$

Computing the mean (for GPs and RVMs) We have, using equation (21)

$$m(\mu_{x^*}, \mathbf{\Sigma}_{x^*}) = E_{\mathbf{x}^*}[\mu(\mathbf{x}^*)] = \int \mu(\mathbf{x}^*) p(\mathbf{x}^*) d\mathbf{x}^* = \sum_j \beta_j \int h(\mathbf{x}^*, \mathbf{x}_j) p(\mathbf{x}^*) d\mathbf{x}^* \tag{26}$$

where $h(\mathbf{x}^*, \mathbf{x}_j) = C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_j)$ is as given by (2) for GPs and $h(\mathbf{x}^*, \mathbf{x}_j) = \phi_j(\mathbf{x}^*)$ is as given by (9) for RVMs.

Let $I_j = \int h(\mathbf{x}^*, \mathbf{x}_j)p(\mathbf{x}^*)d\mathbf{x}^*$. To easily solve the integral, the "trick" consists in writing $h(\mathbf{x}^*, \mathbf{x}_j)$ as a probability density ...

$$I_j = \frac{(2\pi)^{-D/2}|\mathbf{\Lambda}|^{-1/2}}{(2\pi)^{-D/2}|\mathbf{\Lambda}|^{-1/2}} \int h(\mathbf{x}^*, \mathbf{x}_j)p(\mathbf{x}^*)d\mathbf{x}^* \tag{27}$$

so that we have $(2\pi)^{-D/2}|\mathbf{\Lambda}|^{-1/2}h(\mathbf{x}^*, \mathbf{x}_j)$ is a normal distribution with mean $\mathbf{x}_j$ and covariance $\mathbf{\Lambda}$. Now, using the formula giving the multiplication of two Gaussian distributions[1] we have $I_j = (2\pi)^{D/2}|\mathbf{\Lambda}|^{1/2}z_c$ with

$$\begin{aligned}
z_c &= (2\pi)^{-D/2}|C|^{1/2}|\mathbf{\Lambda}|^{-1/2}|\mathbf{\Sigma}_{\mathbf{x}^*}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_j{}^T\mathbf{\Lambda}^{-1}\mathbf{x}_j + \mu_{\mathbf{x}^*}^T\mathbf{\Sigma}_{\mathbf{x}^*}^{-1}\mu_{\mathbf{x}^*} - c_j^T C^{-1}c_j)\right] \\
C &= (\mathbf{\Lambda}^{-1} + \mathbf{\Sigma}_{\mathbf{x}^*}^{-1})^{-1} \\
c_j &= C(\mathbf{\Lambda}^{-1}\mathbf{x}_j + \mathbf{\Sigma}_{\mathbf{x}^*}^{-1}\mu_{\mathbf{x}^*})
\end{aligned} \tag{28}$$

Then, I_j simplifies into[2]

$$I_j = |\mathbf{\Lambda}^{-1}\mathbf{\Sigma}_{\mathbf{x}^*} + I|^{-1/2}\exp\left(-\frac{1}{2}(\mu_{\mathbf{x}^*} - \mathbf{x}_j)^T(\mathbf{\Sigma}_{\mathbf{x}^*} + \mathbf{\Lambda})^{-1}(\mu_{\mathbf{x}^*} - \mathbf{x}_j)\right) \tag{29}$$

where I is the $D \times D$ identity matrix.

We can therefore write the mean as

$$m(\mu_{x^*}, \mathbf{\Sigma}_{x^*}) = \sum_j \beta_j I_j \tag{30}$$

Computing the variance for GPs We have $v(\mu_{\mathbf{x}^*}, \mathbf{\Sigma}_{x^*}) = E_{\mathbf{x}^*}[\sigma^2(\mathbf{x}^*)] + \mathrm{var}_{\mathbf{x}^*}(\mu(\mathbf{x}^*)) = E_{\mathbf{x}^*}[\sigma^2(\mathbf{x}^*)] + E_{\mathbf{x}^*}[\mu(\mathbf{x}^*)^2] - E^2_{\mathbf{x}^*}[\mu(\mathbf{x}^*)]$, which translates into

$$\begin{aligned}
v(\mu_{\mathbf{x}^*}, \mathbf{\Sigma}_{x^*}) = \int \left(C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}^*) - \sum_i\sum_j C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_i)\mathbf{K}_{ij}{}^{-1}C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_j)\right)p(\mathbf{x}^*)d\mathbf{x}^* + \\
\sum_i\sum_j \beta_i\beta_j \int C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_i)C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_j)p(\mathbf{x}^*)d\mathbf{x}^* - \left[\sum_j \beta_j \int C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}_j)p(\mathbf{x}^*)d\mathbf{x}^*\right]^2
\end{aligned} \tag{31}$$

which simplifies into

$$v(\mu_{\mathbf{x}^*}, \mathbf{\Sigma}_{x^*}) = C_{\mathrm{GP}}(\mathbf{x}^*, \mathbf{x}^*) - \sum_i\sum_j(\mathbf{K}_{ij}{}^{-1} - \beta_i\beta_j)I_{ij} - [\sum_j \beta_j I_j]^2 \tag{32}$$

---

[1] $\mathcal{N}(a, A)\mathcal{N}(b, B) \propto \mathcal{N}(c, C)$ with $C = (A^{-1} + B^{-1})^{-1}$, $c = C(A^{-1}a + B^{-1}b)$ and normalising constant $z_c = (2\pi)^{-D/2}|C|^{1/2}|A|^{-1/2}|B|^{-1/2}\exp\left[-\frac{1}{2}(a^T A^{-1}a + b^T B^{-1}b - c^T C^{-1}c)\right]$.

[2] We use here the following identities:

$$\begin{aligned}
(\mathbf{\Sigma}_{\mathbf{x}^*} + \mathbf{\Lambda})^{-1} &= \mathbf{\Sigma}_{\mathbf{x}^*}^{-1} - \mathbf{\Sigma}_{\mathbf{x}^*}^{-1}(\mathbf{\Sigma}_{\mathbf{x}^*}^{-1} + \mathbf{\Lambda}^{-1})^{-1}\mathbf{\Sigma}_{\mathbf{x}^*}^{-1} = \mathbf{\Lambda}^{-1} - \mathbf{\Lambda}^{-1}(\mathbf{\Sigma}_{\mathbf{x}^*}^{-1} + \mathbf{\Lambda}^{-1})^{-1}\mathbf{\Lambda}^{-1} \\
(\mathbf{\Sigma}_{\mathbf{x}^*} + \mathbf{\Lambda})^{-1} &= \mathbf{\Sigma}_{\mathbf{x}^*}^{-1}(\mathbf{\Sigma}_{\mathbf{x}^*}^{-1} + \mathbf{\Lambda}^{-1})^{-1}\mathbf{\Lambda}^{-1} = \mathbf{\Lambda}^{-1}(\mathbf{\Sigma}_{\mathbf{x}^*}^{-1} + \mathbf{\Lambda}^{-1})^{-1}\mathbf{\Sigma}_{\mathbf{x}^*}^{-1}
\end{aligned}$$

where I_j is given by (29) and I_{ij} is as follows (making use of the product of three Gaussians)

$$
\begin{aligned}
I_{ij} = |2\mathbf{\Lambda}^{-1}\mathbf{\Sigma}_{\mathbf{x}^*} + I|^{-1/2} \exp\left(-\frac{1}{2}\mu_{\mathbf{x}^*}^T \mathbf{\Sigma}_{\mathbf{x}^*}^{-1} \mu_{\mathbf{x}^*}\right) \\
\exp\left(-\frac{1}{2}\mathbf{x}_j^T \mathbf{\Lambda}^{-1}\mathbf{x}_j\right) \exp\left(-\frac{1}{2}\mathbf{x}_i^T \mathbf{\Lambda}^{-1}\mathbf{x}_i\right) \exp\left(\frac{1}{2}d_{ij}^T D^{-1} d_{ij}\right)
\end{aligned}
$$

with

$$
D = (2\mathbf{\Lambda}^{-1} + \mathbf{\Sigma}_{\mathbf{x}^*}^{-1})^{-1} \tag{33}
$$

$$
d_{ij} = D(\mathbf{\Lambda}^{-1}(\mathbf{x}_j + \mathbf{x}_i) + \mathbf{\Sigma}_{\mathbf{x}^*}^{-1}\mu_{\mathbf{x}^*}) \tag{34}
$$

which simplifies into:

$$
I_{ij} = |2\mathbf{\Lambda}^{-1}\mathbf{\Sigma}_{\mathbf{x}^*} + I|^{-1/2} \exp\left(-\frac{1}{2}\left[(\mu_{\mathbf{x}^*} - \mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{\Sigma}_{\mathbf{x}^*} + 2\mathbf{\Lambda})^{-1}(\mu_{\mathbf{x}^*} - \mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T\mathbf{\Lambda}^{-1}\mathbf{x}_j\right]\right) \tag{35}
$$

Computing the variance for RVMs We have $v(\mu_{\mathbf{x}^*}, \mathbf{\Sigma}_{x^*}) = E_{\mathbf{x}^*}[\sigma^2(\mathbf{x}^*)] + \mathrm{var}_{\mathbf{x}^*}(\mu(\mathbf{x}^*)) = E_{\mathbf{x}^*}[\sigma^2(\mathbf{x}^*)] + E_{\mathbf{x}^*}[\mu(\mathbf{x}^*)^2] - E^2_{\mathbf{x}^*}[\mu(\mathbf{x}^*)]$, which for RVMs, using (14) translates into

$$
\begin{aligned}
v(\mu_{\mathbf{x}^*}, \mathbf{\Sigma}_{x^*}) = \int \sum_i \sum_j \mathbf{\Sigma}_{ij}{}^{-1}\phi_i(\mathbf{x}^*)\phi_j(\mathbf{x}^*)p(\mathbf{x}^*)d\mathbf{x}^* + \\
\sum_i \sum_j \omega_i\omega_j \int \phi_i(\mathbf{x}^*)\phi_j(\mathbf{x}^*)p(\mathbf{x}^*)d\mathbf{x}^* - \left[\sum_j \omega_j \int \phi_j(\mathbf{x}^*)p(\mathbf{x}^*)d\mathbf{x}^*\right]^2
\end{aligned} \tag{36}
$$

which simplifies into

$$
v(\mu_{\mathbf{x}^*}, \mathbf{\Sigma}_{x^*}) = \delta_{\text{bias}}\left(\mathbf{\Sigma}_{00}^{-1} + 2\sum_i \mathbf{\Sigma}_{0i}^{-1} I_i\right) + \sum_i \sum_j (\mathbf{\Sigma}_{ij}{}^{-1} + \omega_i\omega_j)I_{ij} - [\sum_j \omega_j I_j]^2 \tag{37}
$$

where δ_{bias} is 0 if there is no bias term, and 1 if there is. Again, I_j and I_{ij} are given respectively by equations (29) and (35). The quantity ω_i is the i-th component of the maximum posterior estimate of the weights, given by (15).

5 Time-Series Forecasting

We wish to apply these results to the multiple-step ahead prediction task of time series. Currently, this can be achieved by either training the model to learn how to predict k steps ahead (direct method) or by making repetitive one-step ahead predictions (iterative method). We are concerned with the iterative approach and suggest to propagate the uncertainty as we predict ahead in time.

5.1 "Naive" iterative k-step ahead prediction

Consider the time series $y_{t_1}, \ldots, y_t$ and the state-space model

$$\begin{cases} x_{t_i} = [y_{t_i-1}, \ldots, y_{t_i-L}]^T \\ y_{t_i} = f(x_{t_i}) + \epsilon_{t_i} \end{cases} \tag{38}$$

where the *state* x at time t_i is composed of previous outputs, up to a given lag[3] L and we have an additive (white) noise with variance σ^2.

The naive iterative k-step ahead prediction method works as follows: it predicts only one time step ahead, using the estimate of the output of the current prediction, as well as previous outputs (up to the lag L), as the input to the prediction of the next time step, until the prediction k steps ahead is made.

Using the model (38) and assuming the data is known up to, say, time step t, the prediction of y at $t + k$ is computed via

$$\begin{array}{rcl} x_{t+1} = [y_t, y_{t-1}, \ldots, y_{t+1-L}]^T & \rightarrow & f(x_{t+1}) \sim \mathcal{N}(\mu(x_{t+1}), \sigma^2(x_{t+1})) \\ & & \hat{y}_{t+1} = \mu(x_{t+1}) \\ x_{t+2} = [\hat{y}_{t+1}, y_t, \ldots, y_{t+2-L}]^T & \rightarrow & f(x_{t+2}) \sim \mathcal{N}(\mu(x_{t+2}), \sigma^2(x_{t+2})) \\ & & \hat{y}_{t+2} = \mu(x_{t+2}) \\ & \vdots & \\ x_{t+k} = [\hat{y}_{t+k-1}, \hat{y}_{t+k-2}, \ldots, \hat{y}_{t+k-L}]^T & \rightarrow & f(x_{t+k}) \sim \mathcal{N}(\mu(x_{t+k}), \sigma^2(x_{t+k})) \\ & & \hat{y}_{t+k} = \mu(x_{t+k}) \end{array}$$

where the point estimates $\mu(x_{t+k-i})$ are computed using equation (4). This setup does not account for the uncertainty induced by each successive prediction (variance $\sigma^2(x_{t+k-i}) + \sigma^2$ associated to each $\hat{y}$, given by (5)).

5.2 Propagating the uncertainty

Using the results derived in the previous section, we propose to formally incorporate the uncertainty information about the future regressor. That is, as we predict ahead in time, we now view the lagged outputs as random variables.

In this framework, if, as before, data are known up to time t and we wish to predict k steps ahead, we now have

- at $t + 1$,

$$x_{t+1} \sim \mathcal{N}\left(\begin{bmatrix} y_t \\ \ldots \\ y_{t+1-L} \end{bmatrix}, \begin{bmatrix} 0 & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & 0 \end{bmatrix} \right)$$

  predict $y_{t+1} \sim \mathcal{N}(m(x_{t+1}), v(x_{t+1}) + \sigma^2)$, using (30) and (32), with $x^* = x_{t+1}$

[3] We are not concerned with the identification of the lag and assume it has a known, fixed value.

- at $t+2$,

$$x_{t+2} \sim \mathcal{N}\left(\begin{bmatrix} m(x_{t+1}) \\ \dots \\ y_{t+2-L} \end{bmatrix}, \begin{bmatrix} v(x_{t+1})+\sigma^2 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{bmatrix}\right)$$

predict $y_{t+2} \sim \mathcal{N}(m(x_{t+2}), v(x_{t+2})+\sigma^2)$

$$\vdots$$

- at $t+k$,

$$x_{t+k} \sim \mathcal{N}\left(\begin{bmatrix} m(x_{t+k-1}) \\ \dots \\ m(x_{t+k-L}) \end{bmatrix}, \begin{bmatrix} v(x_{t+k-1})+\sigma^2 & \dots & \mathrm{cov}(y_{t+k-1}, y_{t+k-L}) \\ \dots & \dots & \dots \\ \mathrm{cov}(y_{t+k-L}, y_{t+k-1}) & \dots & v(x_{t+k-L}+\sigma^2) \end{bmatrix}\right)$$

predict $y_{t+k} \sim \mathcal{N}(m(x_{t+k}), v(x_{t+k})+\sigma^2)$

5.2.1 Input distribution

At time t_k, we have the random input vector $x_{t_k} = [y_{t_k-1}, \dots, y_{t_k-L}]^T$ with mean formed of the predicted means of the lagged outputs $y_{t_k-\tau}$, $\tau = 1, \dots, L$, given by (30).

The $L \times L$ input covariance matrix has the different predicted variances on its diagonal: $[\Sigma_{x_{t_k}}]_{ii} = v(x_{t_k-i})$, for $i = 1 \dots L$, computed with (32).

The cross-covariance terms are obtained as follows: at time step t_k, we predict y_{t_k} and then, for the next time step, we need to compute the covariance between y_{t_k} and $[y_{t_k-1}, \dots, y_{t_k+1-L}]$. That is, in general, we want to compute $\mathrm{cov}(y_{t_k}, x_{t_k})$. That is

$$\mathrm{cov}(y_{t_k}, x_{t_k}) = E[y_{t_k} x_{t_k}] - E[y_{t_k}]E[x_{t_k}] \tag{39}$$

with $E[y_{t_k}]$ given by (30) and $E[x_{t_k}] = \mu_{x_{t_k}}$.

We have $E[y_{t_k} x_{t_k}] = \int\int y_{t_k} x_{t_k} p(y_{t_k} x_{t_k}) dy_{t_k} dx_{t_k} = \int\int y_{t_k} x_{t_k} p(y_{t_k}|x_{t_k}) p(x_{t_k}) dy_{t_k} dx_{t_k}$ which is $\int x_{t_k} \mu(x_{t_k}) p(x_{t_k}) dx_{t_k}$ with $\mu(x_{t_k})$ as given by (4).

Replacing and solving this integral in a similar way to what we did for the calculation of the mean, we arrive at

$$E[y_{t_k} x_{t_k}] = \sum_j \beta_j I_j c_j \tag{40}$$

So that the cross-covariance terms are given by

$$\mathrm{cov}(y_{t_k}, x_{t_k}) = \sum_j \beta_j I_j (c_j - \mu_{x_{t_k}}) \tag{41}$$

where I_j and c_j are given by (29) and (28) respectively.

6 When not all components of $\mathbf{x}^*$ are stochastic

References

[Girard et al., 2002] Girard, A., Rasmussen, C. E., and Murray-Smith, R. (2002). Gaussian process priors with uncertain inputs: Multiple-step ahead prediction. Technical report, Department of Computing Science, Glasgow University. http://www.dcs.gla.ac.uk/~agathe/reports.html.

[Girard et al., 2003] Girard, A., Rasmussen, C. E., and Murray-Smith, R. (2003). Gaussian process with uncertain input - application to multiple-step ahead time-series forecasting. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*. MIT Press.

[MacKay, 1997] MacKay, D. J. C. (1997). Gaussian processes - a replacement for supervised neural networks? Lecture notes for a tutorial in Advances in Neural Information Processing Systems.

[Tipping, 2001] Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244.