Introduction to Gaussian Processes

Filipe P. Farias
Teleinformatics Engineering Department
Federal University of Ceará
filipepfarias@fisica.ufc.br

Abstract

A wide variety of methods exists to deal with supervised learning, as restrict a class of linear functions of the inputs, as linear regression, or give a prior probability to every possible function, giving high probability to the functions we consider more likely. The second approach is a way to Gaussian process itself. We will make the pathway through a intuitive construction of this framework.

1 Introduction

The problem of searching for patterns in data is a fundamental one and has a long and successful history. The discovery of regularities in atomic spectra played a key role in the development and verification of quantum physics in the early twentieth century. The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories.

Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as *supervised learning* problems. If the desired output consists of one or more continuous variables, then the task is called *regression*. An example of a regression problem would be the prediction of the yield in a chemical manufacturing process in which the inputs consist of the concentrations of reactants, the temperature, and the pressure. [Bishop, 2006]

In general we denote the input as $\mathbf{x}$, and the output (or target) as y. The input is usually represented as a vector $\mathbf{x}$ as there are in general many input variables. We have a dataset $\mathcal{D}$ of n observations, $\mathcal{D} = \{(\mathbf{x}_i, t_i)\,|i = 1, \ldots, n\}$. Given this training data we wish to make predictions for new inputs $\mathbf{x}^*$ that we have not seen in the *training set*. Thus it is clear that the problem at hand is inductive; we need to move from the finite training data $\mathcal{D}$ to a function f that makes predictions for all possible input values. To do this we must make assumptions about the characteristics of the underlying function, as otherwise any function which is consistent with the training data would be equally valid.

A wide variety of methods have been proposed to deal with the *supervised learning* problem; here we describe two common approaches. The first is to restrict the class of functions that we consider, for example by only considering linear functions of the input. The second approach is (speaking rather loosely) to give a prior probability to every possible function, where higher probabilities are given to functions that we consider to be *more likely*, for example because they are smoother than other functions.

The first approach has an obvious problem in that we have to decide upon the richness of the class of functions considered; if we are using a model based on a certain class of functions (e.g. linear functions) and the target function is not well modelled by this class, then the predictions will be poor. One may be tempted to increase the flexibility of the class of functions, but this runs into the danger of *overfitting*, where we can obtain a good fit to the training data, but perform badly when making test predictions.

The second approach appears to have a serious problem, in that surely there are an uncountably infinite set of possible functions, and how are we going to compute with this set in finite time? This is where the Gaussian *process* comes to our rescue. A Gaussian process is a generalization of the Gaussian probability *distribution*. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic *process* governs the properties of functions. Leaving mathematical sophistication aside, one can loosely think of a function as a very long vector, each entry in the vector specifying the function value $f(x)$ at a particular input x. It turns out, that although this idea is a little naive, it is surprisingly close what we need. Indeed, the question of how we deal computationally with these infinite dimensional objects has the most pleasant resolution imaginable: if you ask only for the properties of the function at a finite number of points, then inference in the Gaussian process will give you the same answer if you ignore the infinitely many other points, as if you would have taken them all into account! And these answers are consistent with answers to any other finite queries you may have. One of the main attractions of the Gaussian process framework is precisely that it unites a sophisticated and consistent view with computational tractability.

2 Linear Regression

Starting with a simple regression problem. Be the data set $\mathcal{D} = \{x_i, t_i | i = 0, \ldots, N-1\}$, where we observe a real-valued input variable x and a measured real-valued variable t. Then, we'll use synthetically generated data for comparison against any learned *model*. And N will be the number of observations of the value t. Our objective is make predictions of the new value $\hat{t}$ for some new input $\hat{x}$.

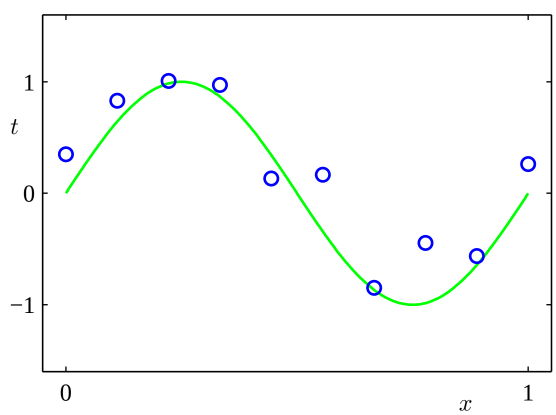

Figure 1: Training data set with $n = 10$ points in blue. The green curve shows the function $sin(2\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x, without knowledge of the green curve.

For this example, we'll use a simple approach based on curve fitting by the polynomial model, i.e., being the function

$$y(x, \mathbf{w}) = \sum_{j=0}^{M-1} w_j x^j \tag{2.1}$$

where M is the order of the polynomial and $\mathbf{w} = [w_0, \ldots, w_M]$ its coefficients. It's important to note that the y isn't linear in x but in $\mathbf{w}$. These functions which are linear on the unknown parameters are called *linear models*.

We can extend the class of models considering linear combinations of nonlinear functions of the input variables, i.e.

$$y(x, \mathbf{w}) = \sum_{j=0}^{M-1} \phi_j(x) w_j \tag{2.2}$$

where $\phi_j(x)$ are known as *basis functions*, and then the total number of parameters for this model will be M. We can evaluate the same operation of (2.2) in the matrix form by

$$y(x, \mathbf{w}) = \boldsymbol{\phi}(x)^\top \mathbf{w} \tag{2.3}$$

where $\boldsymbol{\phi}(x) = [\phi_0(x), \ldots, \phi_{M-1}(x)]^\top$. In the example of the curve fitting, the polynomial regression implies that $\phi_j(x) = x^j$. It's important to note that these linear models are needed to define its basis functions before the training data set is observed.

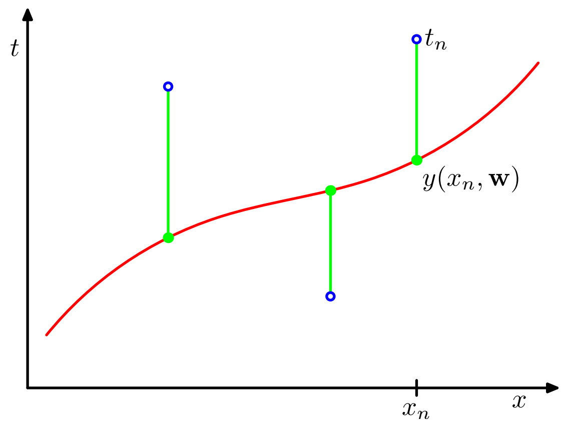

Figure 2: The error function (2.4) corresponds to (one half of) the sum of the squares of the displacements (shown by the vertical green bars) of each data point from the function $y(x, \mathbf{w})$.

The values of $\mathbf{w}$ are obtained by minimizing the *error function*, a measure of the distance between the training data set and y, given values of $\mathbf{w}$. By the way, the chosen error function will be

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_i, \mathbf{w}) - y_i\}^2 \tag{2.4}$$

This indicate that if E is zero, y passes exactly through each training data point. Observe that E do not assume negative values because of its quadratic form, then we can find $\mathbf{w}$ by finding the minimum value of E, denoted $\mathbf{w}^*$, by

$$\frac{\partial E}{\partial \mathbf{w}} = 0 \tag{2.5}$$

We can rewrite (2.4) in the matrix form, as $\mathbf{y} = y(\mathbf{x}, \mathbf{w})$, where $\mathbf{x} = [x_0, \ldots, x_{N-1}]^\top$, i.e. y evaluated for all input variables (See Appendix A.1). Then we have

$$\mathbf{y} = \Phi \mathbf{w} \tag{2.6}$$

where Φ is the *design matrix* such that $\boldsymbol{\phi}(x)$ is evaluated for all $\mathbf{x}$. Proceeding with the minimization, we obtain the optimal $\mathbf{w}$, or $\mathbf{w}^*$ by

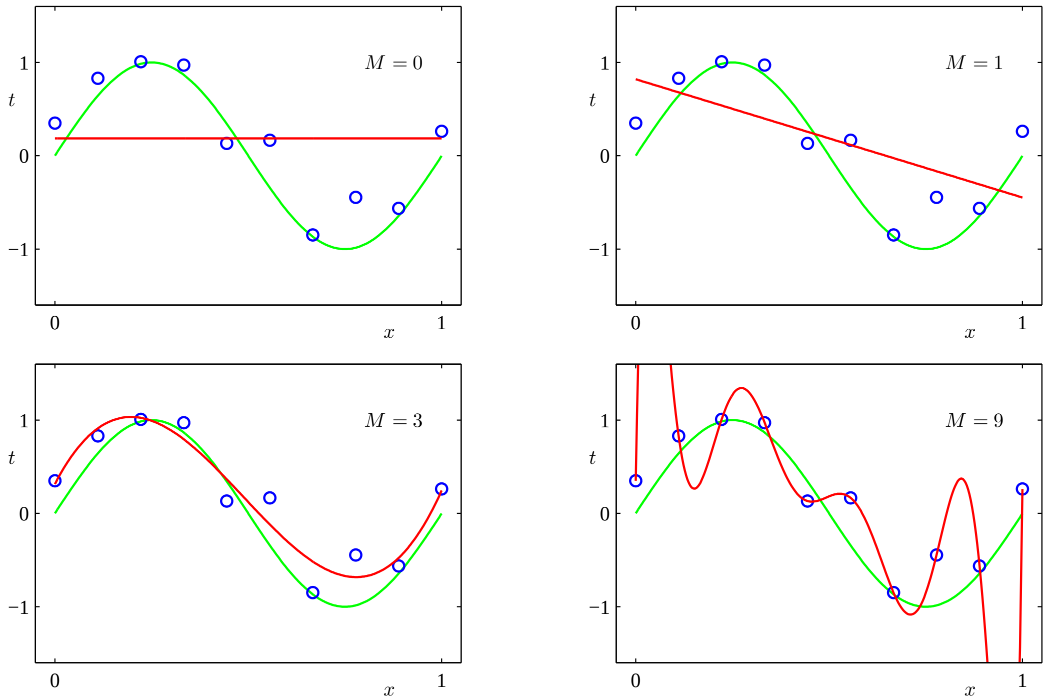

Figure 3: Plots of polynomials for the model in (2.2) having various orders M, shown as red curves.

$$\mathbf{w}^* = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{t} \tag{2.7}$$

which are the parameters that best fit the model to the data.

As we increase the number of parameters, our model becomes more flexible and then our error function approximates of zero for the training data. But when compared to the test data, the error increases. This is known as *over-fitting* as seen in Figure 3 for $M = 9$.

2.1 Regularized Linear Regression

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$\mathbf{w}_0^*$	0.19	0.82	0.31	0.35
$\mathbf{w}_1^*$		-1.27	7.99	232.37
$\mathbf{w}_2^*$			-25.43	-5321.83
$\mathbf{w}_3^*$			17.37	48568.31
$\mathbf{w}_4^*$				-231639.30
$\mathbf{w}_5^*$				640042.26
$\mathbf{w}_6^*$				-1061800.52
$\mathbf{w}_7^*$				1042400.18
$\mathbf{w}_8^*$				-557682.99
$\mathbf{w}_9^*$				125201.43

Table 1: Table of the coefficients $\mathbf{w}^*$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

An approach to minimize the over-fitting problem is to control the flexibility of the model. In the Table 1 we can note that for a larger number of parameters, if we take the derivative of y. Then to control the over-fitting, we can be done by controlling the norm of $\mathbf{w}^*$ as the number of parameters increases. By (2.4) we can add the penalty term $||\mathbf{w}||^2$ scaled by the factor $\lambda/2$, then

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_i,\mathbf{w}) - y_i\}^2 + \frac{\lambda}{2}||\mathbf{w}||^2 \tag{2.8}$$

whats means that our error increases as the norm of $\mathbf{w}$ grows. This will lead us to the matrix form

$$\mathbf{w}^* = (\Phi^\top\Phi + \lambda\mathbf{I})^{-1}\Phi^\top\mathbf{t} \tag{2.9}$$

This allow us to increase the number of parameters trying to control the over-fitting. More, add parameters will allows us to capture different aspects of the data set, what we will see later.

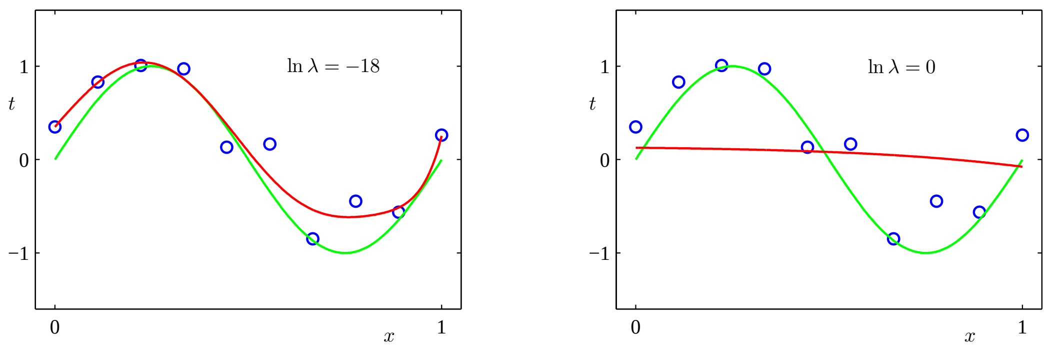

Figure 4: Plots of polynomials for the model in (2.2) having various orders M, shown as red curves.

3 Bayesian Linear Regression

3.1 A Bayesian view of Linear Regression

Until now, we see the curve fitting problem in terms of error minimization. Then we will see the same by a probabilistic perspective gaining some insights into error minimization and regularization, leading us to a full Bayesian treatment.

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} \tag{3.1}$$

We can use the Bayes' theorem (3.1) to convert a *prior* probability into a *posterior* probability at the light of some evidence. We can make inferences about quantities such as the parameters $\mathbf{w}$ in the form of a prior distribution $p(\mathbf{w})$. The observation of the data $\mathcal{D}$ and what its implies in the parameters is expressed as a conditional probability $p(\mathcal{D}|\mathbf{w})$. Then we can evaluate the uncertainty about $\mathbf{w}$ after observed the data $\mathcal{D}$ as a posterior probability $p(\mathbf{w}|\mathcal{D})$.

The quantity $p(\mathcal{D}|\mathbf{w})$ expresses how probable the observed data $\mathcal{D}$ is for different settings of $\mathbf{w}$. Then, not being a probability distribution over the parameters, its integral with respect to $\mathbf{w}$ could not be equal one, then to normalize the equation with respect to the left-side there's a term $p(\mathcal{D})$. This distribution is called *likelihood function*.

Integrating the both sides with respect to $\mathbf{w}$, we obtain the denominator, then considering that integrating a probability distribution over itself is equal to one, we have

$$p(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w} \tag{3.2}$$

3.2 Bayesian curve fitting

Let's consider the same data set $\mathcal{D}$ presented before, but now we have some uncertainty over the value of the measured value t. This uncertainty can be represented as a probability distribution p, in this particular case a Gaussian distribution, with a mean equal to the model $y(x, \mathbf{w})$. Thus we have

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right) \tag{3.3}$$

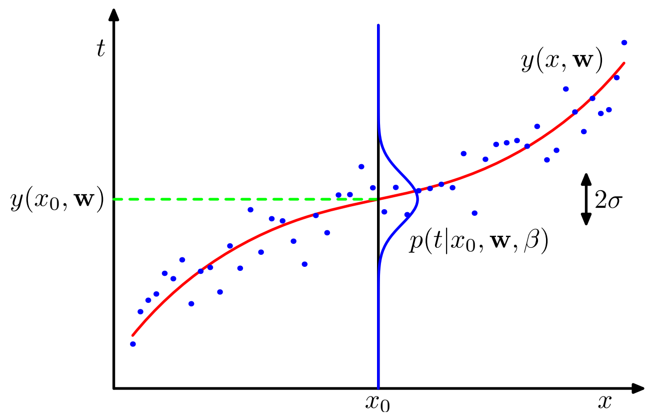

Figure 5: Schematic illustration of a Gaussian conditional distribution for t given x given by (3.3), in which the mean is given by the polynomial function y(x, w), and the precision is given by the parameter β, which is related to the variance $\beta^{-1} = \sigma^2$.

Where β is the variance of the distribution. Note that a large β will give is more uncertainty about the measured value t, then we can call it of *precision parameter*, i.e. how much certain we are about t. As we done in linear regression, we are trying to obtain the parameters for the model. In other words, given a value t, we trying to obtain the *mean* and the *variance* which maximize the probability of the measured value. Assuming the data set being independent and identically distributed, the joint probability of the whole data set will be

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{i=0}^{N-1} \mathcal{N}\left(y_i|y(x_i, \mathbf{w}), \beta^{-1}\right) \tag{3.4}$$

When viewed as function of $y(x_i, \mathbf{w})$, the model, and β^{-1}, this is the likelihood function for the Gaussian. The parameters of the distribution can be determined by maximizing the likelihood function. It is convenient to maximize the log of the likelihood function, or minimize the negative log whats is equivalent, this implies that the maximization of the log of the function is equivalent to the maximization of the function itself, because the logarithm is a monotonically increasing of its argument. This helps the mathematical analysis and helps numerically because the small probabilities can easily underflow the numerical precision of the computer. Then†

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{i=1}^{N}\left\{y\left(x_i, \mathbf{w}\right) - t_i\right\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) \tag{3.5}$$

†Consider the Gaussian distribution as $\mathcal{N}\left(x|\mu, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-1/2}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$

The maximization of (3.5) taking the derivative with respect to $\mathbf{w}$ will lead us back to the same of the minimization of (2.4), the error function of the linear regression. Here, just by notation, we will call the resulting parameters of the maximization of $\mathbf{w}_{\text{ML}}$, what it is called *maximum likelihood*.

We can determine the precision parameter using the maximum likelihood by taking the derivative with respect to β of (3.5), what gives

$$\frac{1}{\beta_{\text{ML}}} = \frac{1}{N}\sum_{i=0}^{N-1}\left\{y\left(x_i, \mathbf{w}_{\text{ML}}\right) - t_i\right\}^2 \tag{3.6}$$

Now we have a probabilistic view of the regression and then we can make predictions for new values of x, given that our model is capable of learn the parameters. And not just one collection of them, but a distribution probability.

In other words, after find the maximum likelihood parameters $\mathbf{w}_{\text{ML}}$ and β_{ML}, we have the parameters distribution by

$$p\left(t|x, \mathbf{w}_{\text{ML}}, \beta_{\text{ML}}\right) = \mathcal{N}\left(t|y\left(x, \mathbf{w}_{\text{ML}}\right), \beta_{\text{ML}}^{-1}\right) \tag{3.7}$$

Aiming to apply a "more Bayesian" approach, we not have yet a prior distribution to make the inference using the Bayes' rule. We can now introduce here the probability distribution over the parameters $p(\mathbf{w})$ as presented in the Section 3.1. The choice is arbitrary, but for this particular case we will consider

$$p(\mathbf{w}|\alpha) = \mathcal{N}\left(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}\right) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^\top\mathbf{w}\right\} \tag{3.8}$$

where α is the variance, or precision parameter, of the distribution and $M+1$ is the number of parameters of the model, i.e. the length of $\mathbf{w}$. We call *hyperparameters* the variables such α who control the model parameters distribution. And now we have by the Bayes' theorem considering that our *posterior* distribution is proportional to the product between the *likelihood function* and the assumed *prior*, as seen in (3.2) then, assuming the observation of the whole data set

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha) \tag{3.9}$$

As done before, we maximize the posterior probability, i.e. find the most probable value given the data by the term $p(\mathbf{w}|\mathbf{x}, \mathbf{t})$ aside of the distribution parameters. This will result a particular choice of $\mathbf{w}$. We call this approach of *maximum posterior*, or MAP. Then taking the negative logarithm

$$\ln p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto \ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) + \ln p(\mathbf{w}|\alpha) \tag{3.10}$$

Then we substitute the probability distributions founded before. Note that the first term in the right side is the error function founded in (3.5). Then, the terms which the minimization depends of $\mathbf{w}$ are

$$\frac{\beta}{2}\sum_{i=1}^{N}\left\{y\left(x_i, \mathbf{w}\right) - t_i\right\}^2 + \frac{\alpha}{2}\mathbf{w}^\top\mathbf{w} \tag{3.11}$$

And we can note the similarity with the regularized linear regression in (2.8) aside of the term λ, what can be founded by $\lambda = \alpha/\beta$. It's important to note that even called maximum posterior, here it was presented a minimization in terms of the negative logarithm, but this equals to the maximization of the positive logarithm. The signal was chosen just for similarity with the error function.

3.3 Bayesian inference

The similarity mentioned before shows that the Bayesian approach comprise even a model training such as the classical regression, as also the control of the over fitting by the regularization. But to say that our model is in fact Bayesian, we might obtain not just a single value, as in MAP, but its distribution. This requires the application the fully Bayes' theorem as (3.1). Then, we have

$$\overbrace{p(\mathbf{w}|\mathbf{t})}^{posterior} = \frac{p(\mathbf{t}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{t})} = \frac{\overbrace{p(\mathbf{t}|\mathbf{w})}^{likelihood}\ \overbrace{p(\mathbf{w})}^{prior}}{\underbrace{\int p(\mathbf{t}|\mathbf{w})p(\mathbf{w})d\mathbf{w}}_{marginal\ distribution}} \tag{3.12}$$

This is called *Bayesian inference.* If we assume that all distributions which we are working are Gaussian, the posterior distribution has closed form. To do that, we make use of the closure under linear transformations, or *affine transformations*, for the Gaussian. For that, we will make use of the corollary below, which theorems are proven in the Appendix B.

Corollary 1. *Being $\mathbf{x}_b$ conditioned on $\mathbf{x}_a$ and Gaussian distributed as*

$$p\left(\mathbf{x}_a\right) = \mathcal{N}\left(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a\right), \quad p\left(\mathbf{x}_b|\mathbf{x}_a\right) = \mathcal{N}\left(\mathbf{x}_b|\mathbf{M}\mathbf{x}_a + \mathbf{d}, \boldsymbol{\Sigma}_{b|a}\right) \tag{3.13}$$

with $\boldsymbol{\mu}_b = \mathbf{M}\boldsymbol{\mu}_a + \mathbf{d}$, $\boldsymbol{\Sigma}_b = \boldsymbol{\Sigma}_{b|a} + \mathbf{M}\boldsymbol{\Sigma}_a\mathbf{M}^\top$, $\mathbf{M}$ a constant matrix and $\mathbf{d}$ a constant vector, both with the appropriate dimensions. Then conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ is given by

$$p\left(\mathbf{x}_a|\mathbf{x}_b\right) = \mathcal{N}\left(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b}\right) \tag{3.14a}$$

with

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\Sigma}_{a|b}\left(\mathbf{M}^\top\boldsymbol{\Sigma}_{b|a}^{-1}\left(\mathbf{x}_b - \mathbf{d}\right) + \boldsymbol{\Sigma}_a^{-1}\boldsymbol{\mu}_a\right) \tag{3.14b}$$

$$\boldsymbol{\Sigma}_{a|b} = \left(\boldsymbol{\Sigma}_a^{-1} + \mathbf{M}^\top\boldsymbol{\Sigma}_{b|a}^{-1}\mathbf{M}\right)^{-1}. \tag{3.14c}$$

Assuming no deviation in the mean, $\mathbf{d} = \mathbf{0}$, and the linear transformation being our design matrix, $\mathbf{M} = \Phi$, we obtain that

$$p(\mathbf{w}|\mathbf{t}, \alpha, \beta) = \mathcal{N}\left(\mathbf{w}|\boldsymbol{\mu}_{\mathbf{w}|\mathbf{t}}, \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{t}}\right) \tag{3.15}$$

being

$$\boldsymbol{\mu}_{\mathbf{w}|\mathbf{t}} = \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{t}}\left(\beta\Phi^\top\mathbf{t} + \boldsymbol{\Sigma}_{\mathbf{w}}^{-1}\boldsymbol{\mu}_{\mathbf{w}}\right), \quad \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{t}} = \left(\boldsymbol{\Sigma}_{\mathbf{w}}^{-1} + \beta\Phi^\top\Phi\right)^{-1} \tag{3.16}$$

assumed the prior distribution defined in (3.8) and the precision matrix $\boldsymbol{\Sigma}_{\mathbf{t}|\mathbf{w}} = \beta^{-1}\mathbf{I}$. Then we have defined

$$\boldsymbol{\mu}_{\mathbf{w}} = \mathbf{0}, \quad \boldsymbol{\Sigma}_{\mathbf{w}} = \alpha^{-1}\mathbf{I} \tag{3.17}$$

3.4 Predictive distribution

In practice, some times it is more valuable the information about t itself than its parameters $\mathbf{w}$. We can make this by evaluating the predictions of t for the new values of x by

$$p(t|\mathbf{t}, \alpha, \beta) = \int p(t|\mathbf{w}, \beta)p(\mathbf{w}|\mathbf{t}, \alpha, \beta)\mathrm{d}\mathbf{w} \tag{3.18}$$

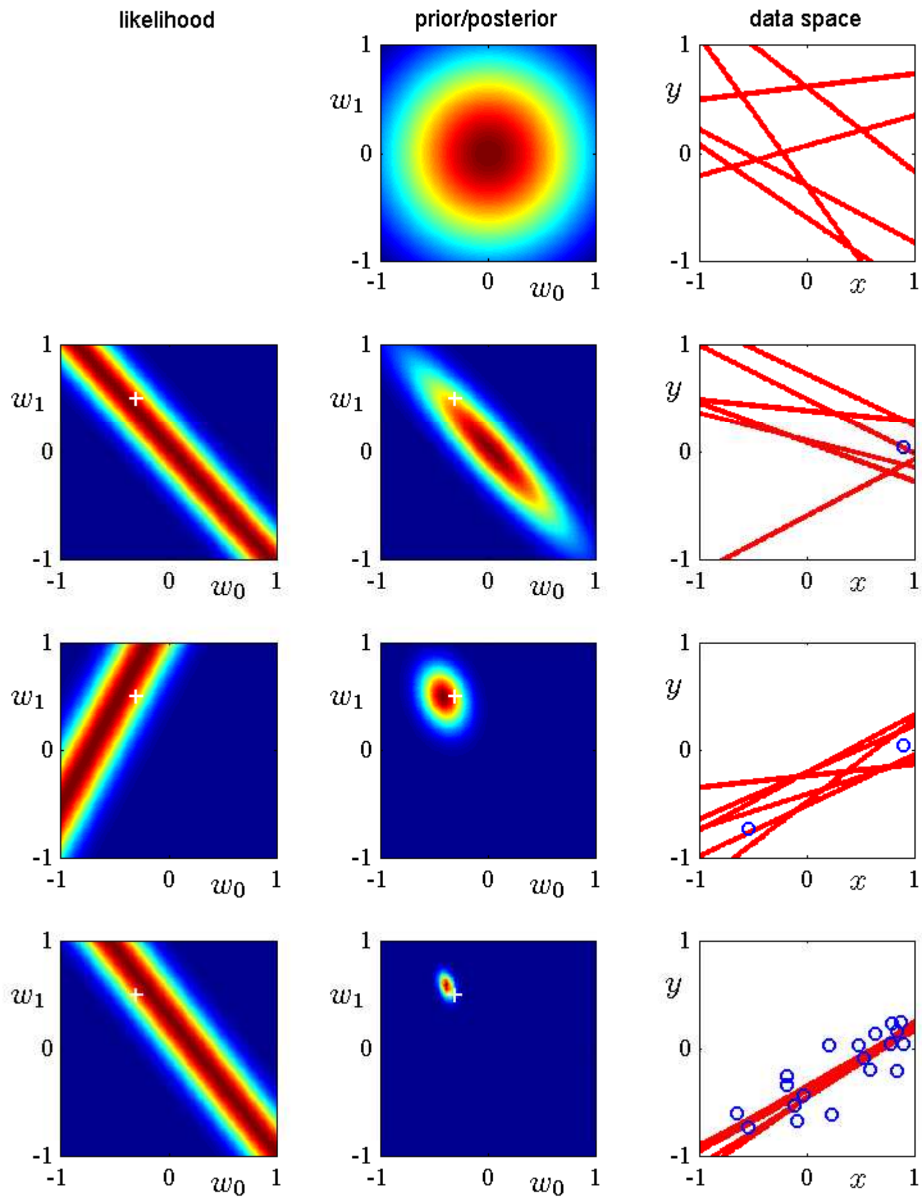

Figure 6: Illustration of sequential Bayesian learning for a simple linear model of the form $y(x, \mathbf{w}) = w_0 + w_1 x$. The hyperparameters α and β are assumed as 2 and 25, respectively, just by example.

what is called *predictive distribution*. The distributions under integration were defined in (3.3) and (3.15). Then we use the *marginalization* defined in Appendix B and obtain that

$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}\left(t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\right) \tag{3.19}$$

with $\boldsymbol{\mu}_y = \boldsymbol{\phi}(x)^\top \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{t}}$ and $\boldsymbol{\Sigma}_y = \beta^{-1} + \boldsymbol{\phi}(x)^\top \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{t}} \boldsymbol{\phi}(x)$. Note that we are considering the linear model as $y(x, \mathbf{w}) = \boldsymbol{\phi}(x)^\top \mathbf{w}$, i.e. the affine transformation $\mathbf{M}$ here is $\boldsymbol{\phi}(x)^\top$.

An alternative formulation [Rasmussen and Williams, 2005] is

$$\begin{aligned} p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}\Big(&\boldsymbol{\phi}(x)^\top \Sigma_{\mathbf{w}} \Phi \left(K + \beta I\right)^{-1} \mathbf{t}\ , \\ &\boldsymbol{\phi}(x)^\top \Sigma_{\mathbf{w}} \boldsymbol{\phi}(x) - \boldsymbol{\phi}(x)^\top \Sigma_{\mathbf{w}} \Phi \left(K + \beta I\right)^{-1} \Phi^\top \Sigma_{\mathbf{w}} \boldsymbol{\phi}(x)\Big) \end{aligned} \tag{3.20}$$

with $K = \Phi^\top \boldsymbol{\Sigma}_{\mathbf{w}} \Phi$.

3.5 Equivalent kernel

Notice that we are using transformations always of the type $\Phi^\top \boldsymbol{\Sigma}_{\mathbf{w}} \Phi$, $\boldsymbol{\phi}(x)^\top \boldsymbol{\Sigma}_{\mathbf{w}} \Phi$, or $\boldsymbol{\phi}(x)^\top \boldsymbol{\Sigma}_{\mathbf{w}} \boldsymbol{\phi}(x)$. Then we can generalize the form $\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_{\mathbf{w}} \boldsymbol{\phi}(\mathbf{x}')$ where $\mathbf{x}$ and $\mathbf{x}'$ are in either the training or the test sets. We define this form as $k(\mathbf{x}, \mathbf{x}')$, where $k(\cdot, \cdot)$ is called *covariance function*, or *kernel*. Further insight into the role of the equivalent kernel can be obtained by considering the covariance between $y(\mathbf{x})$ and $y(\mathbf{x}')$, which is given by

$$\begin{aligned} \operatorname{cov}\left\{y(\mathbf{x}), y(\mathbf{x}')\right\} &= \operatorname{cov}\left\{\boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}, \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})\right\} \\ &= \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_{\mathbf{w}} \boldsymbol{\phi}(\mathbf{x}) = \beta^{-1} k(\mathbf{x}, \mathbf{x}') \end{aligned} \tag{3.21}$$

From the form of the equivalent kernel, we see that the predictive mean at nearby points will be highly correlated, whereas for more distant pairs of points the correlation will be smaller.

4 Gaussian processes

Until now we have made the inference in the *feature space*, the space where the parameters $\mathbf{w}$ are. In other words, the strategy is to train our model, obtaining the parameters probability distribution, by Bayesian inference, and then evaluating the predictive distribution with the posterior of the inference.

An alternative and equivalent way to achieve such results is to make the inference directly in the space of functions, or *function space*. To this we use the *Gaussian processes* to describe the distribution over the functions directly [Rasmussen and Williams, 2005].

We define a Gaussian process (GP) as a collection of random variables, such that any finite number of which is normal jointly distributed. As the normal distribution, the GP is completely defined by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ of a real process $y(\mathbf{x})$, these in turn are defined as

$$m(\mathbf{x}) = \mathbb{E}\left\{y(\mathbf{x})\right\}, \quad k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left\{\left(y(\mathbf{x}) - m(\mathbf{x})\right)\left(y(\mathbf{x}') - m(\mathbf{x}')\right)\right\} \tag{4.1}$$

and finally

$$y(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \tag{4.2}$$

Such collection definition automatically implies in the marginalization property already present for the multidimensional Gaussian distributions. For the GP this means that the observation of a larger set of variables does not change the distribution of the smaller set.

This is important to obtain our Bayesian linear regression as a GP. Being our model $y(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}$ with prior $\mathcal{N}\left(\mathbf{w}|\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}\right)$. We obtain for the mean and covariance functions

$$\begin{aligned} \mathbb{E}\left\{y(\mathbf{x})\right\} &= \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}\left\{\mathbf{w}\right\} = 0, \\ \mathbb{E}\left\{y(\mathbf{x}) y(\mathbf{x}')\right\} &= \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}\left\{\mathbf{w}\mathbf{w}^\top\right\} \boldsymbol{\phi}(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_{\mathbf{w}} \boldsymbol{\phi}(\mathbf{x}') \end{aligned} \tag{4.3}$$

4.1 Prediction with Noisy Observations

The last sections give us an insight about the construction of the learning process for the GP. Analogous to the Bayes' theorem for the Gaussian, we use the idea of the GP as a multidimensional Gaussian distribution and we can make inference with its partitions.

First, to take the concept, we will consider the case where the observations are noise free. We will substitute the covariance matrices from the partitioned Gaussian distributions by the covariance function applied at the points of the observations $\mathbf{y}$ and the prediction $\mathbf{y}_*$, as

$$\mathcal{N}\left(\left(\begin{array}{c}\mathbf{x}_a\\ \mathbf{x}_b\end{array}\right)\middle|\mathbf{0},\left(\begin{array}{cc}\mathbf{K}(\mathbf{x},\mathbf{x}) & \mathbf{K}(\mathbf{x},\mathbf{x}_*)\\ \mathbf{K}(\mathbf{x}_*,\mathbf{x}) & \mathbf{K}(\mathbf{x}_*,\mathbf{x}_*)\end{array}\right)\right) \tag{4.4}$$

where $\mathbf{K}(\cdot,\cdot)$ denotes the covariance matrices evaluated at all pairs of N-dimensional $\mathbf{x}$ training points and N_*-dimensional $\mathbf{x}_*$ test points. Then, making use of Appendix B, we use the *conditioning* to obtain the predictive distribution

$$\begin{aligned}p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{x},\mathbf{y}) = \mathcal{N}\big(&\mathbf{y}_*|\mathbf{K}(\mathbf{x}_*,\mathbf{x})\mathbf{K}(\mathbf{x},\mathbf{x})^{-1}\mathbf{y},\\ &\mathbf{K}(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{K}(\mathbf{x}_*,\mathbf{x})\mathbf{K}(\mathbf{x},\mathbf{x})^{-1}\mathbf{K}(\mathbf{x},\mathbf{x}_*)\big)\end{aligned} \tag{4.5}$$

Now assuming the noise in the observations, what is a more realistic modelling situation, we have that $t = y(\mathbf{x}) + \varepsilon$, being ε the Gaussian noise with variance β^{-1}. Then we have that $\mathrm{cov}(\mathbf{t}) = \mathbf{K}(\mathbf{x},\mathbf{x}) + \beta^{-1}\mathbf{I}$.

Deriving the conditional distribution corresponding we arrive at the key predictive equations for Gaussian process regression

$$\begin{aligned}p(\mathbf{y}_*|\mathbf{x},\mathbf{t},\mathbf{x}_*) &= \mathcal{N}\left(\overline{\mathbf{y}}_*, \mathrm{cov}\left(\mathbf{y}_*\right)\right), \text{ where}\\ \overline{\mathbf{y}}_* &\triangleq \mathbb{E}\left\{\mathbf{y}_*|\mathbf{x},\mathbf{t},\mathbf{x}_*\right\} = \mathbf{K}\left(\mathbf{x}_*,\mathbf{x}\right)\left(\mathbf{K}(\mathbf{x},\mathbf{x}) + \beta^{-1}\mathbf{I}\right)^{-1}\mathbf{t}\\ \mathrm{cov}\left(\mathbf{y}_*\right) &= \mathbf{K}\left(\mathbf{x}_*,\mathbf{x}_*\right) - \mathbf{K}\left(\mathbf{x}_*,\mathbf{x}\right)\left(\mathbf{K}(\mathbf{x},\mathbf{x}) + \beta^{-1}\mathbf{I}\right)^{-1}\mathbf{K}\left(\mathbf{x},\mathbf{x}_*\right)\end{aligned} \tag{4.6}$$

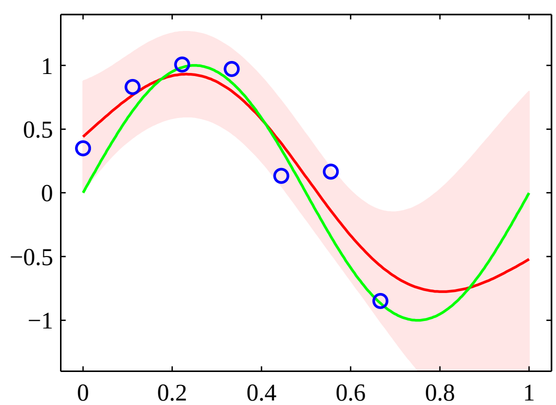

Figure 7: Illustration of Gaussian process regression applied to the sinusoidal data set in which the three right-most data points have been omitted. The green curve shows the sinusoidal function from which the data points, shown in blue, are obtained by sampling and addition of Gaussian noise. The red line shows the mean of the Gaussian process predictive distribution, and the shaded region corresponds to plus and minus two standard deviations. Notice how the uncertainty increases in the region to the right of the data points.

Appendix

A Derivations

A.1 Matrix Form

Be the linear model $f(x, \mathbf{w}) = \boldsymbol{\phi}(x)^\top \mathbf{w}$. Suppose $\Phi = [\boldsymbol{\phi}(x_1), \ldots, \boldsymbol{\phi}(x_N)]^\top$, then Φ will be of the form

$$\Phi = \begin{bmatrix} \phi_0(x_0) & \ldots & \phi_{M-1}(x_0) \\ \vdots & \ddots & \vdots \\ \phi_0(x_{N-1}) & \ldots & \phi_{M-1}(x_{N-1}) \end{bmatrix} \tag{A.1}$$

called *design matrix*. Then the model turns to $\mathbf{f} = \Phi\mathbf{w}$. This will lead us to the matrix form for the quadratic error function

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}(\mathbf{f} - \mathbf{y})^\top(\mathbf{f} - \mathbf{y}) \\ &= \frac{1}{2}(\Phi\mathbf{w} - \mathbf{y})^\top(\Phi\mathbf{w} - \mathbf{y}) \\ &= \frac{1}{2}(\mathbf{w}^\top\Phi^\top\Phi\mathbf{w} - \mathbf{y}^\top\Phi\mathbf{w} - \mathbf{w}^\top\Phi^\top\mathbf{y} + \mathbf{y}^\top\mathbf{y}) \end{aligned}$$

Observe that even in the matrix form, the error function remains scalar, which implies that $\mathbf{y}^\top\Phi\mathbf{w} = \mathbf{w}^\top\Phi^\top\mathbf{y}$ by the transpose of the product rule. Then

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w}^\top\Phi^\top\Phi\mathbf{w} - 2\mathbf{y}^\top\Phi\mathbf{w} + \mathbf{y}^\top\mathbf{y})$$

Then we proceed by the minimization by $\frac{\partial E}{\partial \mathbf{w}} = 0$

$$\begin{aligned} 0 &= \frac{1}{2}(2\mathbf{w}^\top\Phi^\top\Phi - 2\mathbf{y}^\top\Phi)^\dagger \\ \mathbf{w}^{*\top} &= \mathbf{y}^\top\Phi(\Phi^\top\Phi)^{-1} \\ \mathbf{w}^* &= (\Phi^\top\Phi)^{-1}\Phi^\top\mathbf{y} \end{aligned} \tag{A.2}$$

For the regularized linear regression, we do $\frac{\lambda}{2}||\mathbf{w}||^2 = \mathbf{w}^\top\mathbf{w}$, then

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}(\mathbf{f} - \mathbf{y})^\top(\mathbf{f} - \mathbf{y}) + \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w} \\ &= \frac{1}{2}(\mathbf{w}^\top\Phi^\top\Phi\mathbf{w} - \mathbf{y}^\top\Phi\mathbf{w} - \mathbf{w}^\top\Phi^\top\mathbf{y} + \mathbf{y}^\top\mathbf{y}) + \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w} \end{aligned}$$

†Using two facts. First, if $\alpha = \mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{x}$, then $\frac{\partial \alpha}{\partial \mathbf{x}} = 2\mathbf{x}^\top\mathbf{A}$, being α scalar. Second, if $\alpha = \mathbf{y}^\mathrm{T}\mathbf{A}\mathbf{x}$, then $\frac{\partial \alpha}{\partial \mathbf{x}} = \mathbf{y}^\mathrm{T}\mathbf{A}$. For both, $\mathbf{A}$ is independent of $\mathbf{x}$ and $\mathbf{y}$ [Graybill, 1983].

And with the minimization we do $\frac{\partial E}{\partial \mathbf{w}} = 0$, then

$$
\begin{aligned}
0 &= \mathbf{w}^\top \Phi^\top \Phi - \mathbf{y}^\top \Phi + \lambda \mathbf{w}^\top \\
\mathbf{w}^{*\top} &= \mathbf{y}^\top \Phi (\Phi^\top \Phi + \lambda \mathbf{I})^{-1} \\
\mathbf{w}^* &= (\Phi^\top \Phi + \lambda \mathbf{I})^{-1} \Phi^\top \mathbf{y}
\end{aligned}
\tag{A.3}
$$

where $\mathbf{I}$ is the identity matrix.

B Bayes' theorem for Gaussian variables [Schön and Lindsten, 2011]

B.1 Partitioned Gaussian distributions

Be $\mathbf{x}$ a n-dimensional vector with a Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the partitioned will be

$$
\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}. \tag{B.1}
$$

Preserved the symmetry $\boldsymbol{\Sigma}^\top = \boldsymbol{\Sigma}$, we say the covariance matrix is positive definite. And be the multivariate Gaussian

$$
\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}} \frac{1}{(\det \boldsymbol{\Sigma})^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} \tag{B.2}
$$

We define too, just for convenience of work, the precision matrix $\boldsymbol{\Lambda}$ by

$$
\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix} \equiv \boldsymbol{\Sigma}^{-1} \tag{B.3}
$$

assuming all matrices have inverses.

Theorem 1 (Marginalization)**.** *Being the random vector* $\mathbf{x}$ *and its partitioned as above, the marginal density* $p(\mathbf{x}_a)$ *is given by*

$$
p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a | \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})
$$

Proof. The marginal density $p(\mathbf{x}_a)$ is obtained by integrating the joint density $p(\mathbf{x}) = p(\mathbf{x}_a, \mathbf{x}_b)$ with relation to $\mathbf{x}_b$

$$
p(\mathbf{x}_a) = \int p(\mathbf{x}_a, \mathbf{x}_b) \, \mathrm{d}\mathbf{x}_b \tag{B.4}
$$

Then we expand the exponential argument of (B.2) for the partitioned Gaussian

$$
-\frac{1}{2} \left(\begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} - \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \right)^\top \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix} \left(\begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} - \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \right) \tag{B.5}
$$

What implies

$$
\begin{aligned}
&-\frac{1}{2}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)^\top \boldsymbol{\Lambda}_{aa}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)-\frac{1}{2}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)^\top \boldsymbol{\Lambda}_{ab}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)\\
&-\frac{1}{2}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)^\top \boldsymbol{\Lambda}_{ba}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)-\frac{1}{2}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)^\top \boldsymbol{\Lambda}_{bb}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)
\end{aligned} \tag{B.6}
$$

Here we make use of the *Schur complement*, in which, being the partitioned matrix

$$
\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}^{-1}+\mathbf{A}^{-1}\mathbf{B}\mathbf{M}^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{M}^{-1} \\ -\mathbf{M}^{-1}\mathbf{C}\mathbf{A}^{-1} & \mathbf{M}^{-1} \end{pmatrix} \tag{B.7}
$$

the quantity $\mathbf{M}^{-1}$ is the *Schur complement* of the left side matrix with respect to $\mathbf{D}$, defined as

$$
\mathbf{M}=\left(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1} \tag{B.8}
$$

This will motivated the term grouping below

$$
\begin{aligned}
&-\frac{1}{2}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)^\top \boldsymbol{\Lambda}_{aa}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)-\frac{1}{2}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)^\top \boldsymbol{\Lambda}_{ab}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)\\
&-\frac{1}{2}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)^\top \boldsymbol{\Lambda}_{ba}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)-\frac{1}{2}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)^\top \boldsymbol{\Lambda}_{bb}\left(\mathbf{x}_b-\boldsymbol{\mu}_b\right)\\
&=-\frac{1}{2}\left(\mathbf{x}_b^\top \boldsymbol{\Lambda}_{bb}\mathbf{x}_b-2\mathbf{x}_b^\top \boldsymbol{\Lambda}_{bb}\left(\boldsymbol{\mu}_b-\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)\right)-2\mathbf{x}_a^\top \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_b\right.\\
&\left.+2\boldsymbol{\mu}_a^\top \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_b+\boldsymbol{\mu}_b^\top \boldsymbol{\Lambda}_{bb}\boldsymbol{\mu}_b+\mathbf{x}_a^\top \boldsymbol{\Lambda}_{aa}\mathbf{x}_a-2\mathbf{x}_a^\top \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a+\boldsymbol{\mu}_a^\top \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a\right)
\end{aligned}
$$

Then we complete the squares resulting independent

$$
\begin{aligned}
&-\frac{1}{2}\left(\mathbf{x}_b-\left(\boldsymbol{\mu}_b-\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)\right)\right)^\top \boldsymbol{\Lambda}_{bb}\left(\mathbf{x}_b-\left(\boldsymbol{\mu}_b-\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)\right)\right)\\
&+\frac{1}{2}\left(\mathbf{x}_a^\top \boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\mathbf{x}_a-2\mathbf{x}_a^\top \boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\boldsymbol{\mu}_a+\boldsymbol{\mu}_a\boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\boldsymbol{\mu}_a\right)\\
&-\frac{1}{2}\left(\mathbf{x}_a^\top \boldsymbol{\Lambda}_{aa}\mathbf{x}_a-2\mathbf{x}_a^\top \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a+\boldsymbol{\mu}_a^\top \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a\right)\\
&=\underbrace{-\frac{1}{2}\left(\mathbf{x}_b-\left(\boldsymbol{\mu}_b-\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)\right)\right)^\top \boldsymbol{\Lambda}_{bb}\left(\mathbf{x}_b-\left(\boldsymbol{\mu}_b-\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)\right)\right)}_{E_1}\\
&\underbrace{-\frac{1}{2}\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)^\top \left(\boldsymbol{\Lambda}_{aa}-\boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba}\right)\left(\mathbf{x}_a-\boldsymbol{\mu}_a\right)}_{E_2}
\end{aligned}
$$

Now, back to the marginalization, we have

$$
p\left(\mathbf{x}_a\right)=\int \frac{1}{(2\pi)^{n/2}}\frac{1}{\left(\det \boldsymbol{\Sigma}\right)^{1/2}}\exp\left\{E_1\right\}\exp\left\{E_2\right\}d\mathbf{x}_b \tag{B.9}
$$

By inspection, we have that, being the integral of the density function equals one and being n_b the dimension of $\mathbf{x}_b$

$$
\int \exp\left\{E_1\right\}d\mathbf{x}_b=(2\pi)^{n_b/2}\left(\det \boldsymbol{\Lambda}_{bb}^{-1}\right)^{1/2} \tag{B.10}
$$

Then, substituting (B.9) in (B.10) we have

$$p(\mathbf{x}_a) = \frac{1}{(2\pi)^{n_a/2}} \frac{\left(\det \boldsymbol{\Lambda}_{bb}^{-1}\right)^{1/2}}{(\det \boldsymbol{\Sigma})^{1/2}} \exp\{E_2\}$$

We will use the determinant property from Schur complement below

$$\det \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} = \det \mathbf{A} \det \left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right) \tag{B.11}$$

Then, substituting the values of $\boldsymbol{\Sigma}$ we have that

$$\det \boldsymbol{\Sigma} = \det \boldsymbol{\Sigma}_{aa} \det \left(\boldsymbol{\Sigma}_{bb} - \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}\boldsymbol{\Sigma}_{ab}\right) \tag{B.12}$$

Using the Schur complement

$$\det \boldsymbol{\Sigma} = \det \boldsymbol{\Sigma}_{aa} \det \boldsymbol{\Lambda}_{bb}^{-1} \tag{B.13}$$

Finally, using the Schur complement again, we obtain that $\boldsymbol{\Lambda}_{aa} - \boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ba} = \boldsymbol{\Sigma}_{aa}$, what concludes the proof, substituting the result in E_2. □

Theorem 2 (Conditioning)**.** *Being the random vector* $\mathbf{x}$ *and its partitioned as above, the conditional density* $p(\mathbf{x}_a|\mathbf{x}_b)$ *is given by*

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}\left(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b}\right)$$

where $\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)$ *and* $\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Lambda}_{aa}^{-1}$.

Proof. By the product rule we have

$$p(\mathbf{x}_a|\mathbf{x}_b) = \frac{p(\mathbf{x})}{p(\mathbf{x}_b)} \tag{B.14}$$

what is by (B.2),then

$$p(\mathbf{x}_a|\mathbf{x}_b) = \sqrt{\frac{\det \boldsymbol{\Sigma}_{bb}}{(2\pi)^{n_a/2} \det \boldsymbol{\Sigma}}} \exp(E) \tag{B.15}$$

where

$$E = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) + \frac{1}{2}(\mathbf{x}_b - \boldsymbol{\mu}_b)^\top \boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \tag{B.16}$$

Similarly to what was done in *marginalization* and using the Schur complement, the result

$$\det \boldsymbol{\Sigma} = \det \boldsymbol{\Sigma}_{bb} \det \left(\boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\right) = \det \boldsymbol{\Sigma}_{bb} \det \boldsymbol{\Lambda}_{bb}^{-1} \tag{B.17}$$

Using that $\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1}$, we expand the term E

$$E = -\frac{1}{2}(\mathbf{x}_a - \boldsymbol{\mu}_a)^\top \boldsymbol{\Lambda}_{aa}(\mathbf{x}_a - \boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_a - \boldsymbol{\mu}_a)^\top \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \tag{B.18}$$

$$-\frac{1}{2}(\mathbf{x}_b - \boldsymbol{\mu}_b)^\top \boldsymbol{\Lambda}_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_b - \boldsymbol{\mu}_b)^\top \left(\boldsymbol{\Lambda}_{bb} - \boldsymbol{\Sigma}_{bb}^{-1}\right)(\mathbf{x}_b - \boldsymbol{\mu}_b) \tag{B.19}$$

Reordering the terms, we'll have

$$E = -\frac{1}{2}\mathbf{x}_a^\top \mathbf{\Lambda}_{aa}\mathbf{x}_a + \mathbf{x}_a^\top \mathbf{\Lambda}_{aa}\left(\boldsymbol{\mu}_a - \mathbf{\Lambda}_{aa}^{-1}\mathbf{\Lambda}_{ab}\left(\mathbf{x}_b - \boldsymbol{\mu}_b\right)\right)$$
$$- \frac{1}{2}\boldsymbol{\mu}_a^\top \mathbf{\Lambda}_{aa}\boldsymbol{\mu}_a + \boldsymbol{\mu}_a^\top \mathbf{\Lambda}_{ab}\left(\mathbf{x}_b - \boldsymbol{\mu}_b\right) - \frac{1}{2}\left(\mathbf{x}_b - \boldsymbol{\mu}_b\right)^\top \mathbf{\Lambda}_{ba}\mathbf{\Lambda}_{aa}^{-1}\mathbf{\Lambda}_{ab}\left(\mathbf{x}_b - \boldsymbol{\mu}_b\right) \tag{B.20}$$

Completing the squares, we obtain

$$E = \left(\mathbf{x}_a - \left(\boldsymbol{\mu}_a - \mathbf{\Lambda}_{aa}^{-1}\mathbf{\Lambda}_{ab}\left(\mathbf{x}_b - \boldsymbol{\mu}_b\right)\right)\right)^\top \mathbf{\Lambda}_{aa}\left(\mathbf{x}_a - \left(\boldsymbol{\mu}_a - \mathbf{\Lambda}_{aa}^{-1}\mathbf{\Lambda}_{ab}\left(\mathbf{x}_b - \boldsymbol{\mu}_b\right)\right)\right) \tag{B.21}$$

Then by inspection we have for the precision matrix $\mathbf{\Lambda}_{aa}$

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \mathbf{\Lambda}_{aa}^{-1}\mathbf{\Lambda}_{ab}\left(\mathbf{x}_b - \boldsymbol{\mu}_b\right) \tag{B.22a}$$

and analogously

$$\boldsymbol{\mu}_{b|a} = \boldsymbol{\mu}_b - \mathbf{\Lambda}_{bb}^{-1}\mathbf{\Lambda}_{ba}\left(\mathbf{x}_a - \boldsymbol{\mu}_a\right) \tag{B.22b}$$

for $\mathbf{\Lambda}_{bb}$. □

B.2 Affine transformation

Making use of the Theorems 1 and 2, we can show that the Gaussian is closed under linear transformations, i.e. an affine transformation of Gaussians results in another Gaussian.

Theorem 1 (Affine transformation)**.** *Assume $\mathbf{x}_a$ and $\mathbf{x}_b$ are Gaussian distributed and $\mathbf{x}_b$ conditioned by $\mathbf{x}_a$, as*

$$p\left(\mathbf{x}_a\right) = \mathcal{N}\left(\mathbf{x}_a | \boldsymbol{\mu}_a, \mathbf{\Sigma}_a\right), \quad p\left(\mathbf{x}_b | \mathbf{x}_a\right) = \mathcal{N}\left(\mathbf{x}_b | \mathbf{M}\mathbf{x}_a + \mathbf{d}, \mathbf{\Sigma}_{b|a}\right) \tag{B.23}$$

where $\mathbf{M}$ is a constant matrix and $\mathbf{d}$ a constant vector, both with the appropriate dimensions. Then the joint distribution is given by

$$p\left(\mathbf{x}_a, \mathbf{x}_b\right) = \mathcal{N}\left(\left(\begin{array}{c}\mathbf{x}_a \\ \mathbf{x}_b\end{array}\right) \middle| \left(\begin{array}{c}\boldsymbol{\mu}_a \\ \mathbf{M}\boldsymbol{\mu}_a + \mathbf{d}\end{array}\right), \mathbf{R}\right), \tag{B.24}$$

being

$$\mathbf{R} = \left(\begin{array}{cc}\mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1}\mathbf{M} + \mathbf{\Sigma}_a^{-1} & -\mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} \\ -\mathbf{\Sigma}_{b|a}^{-1}\mathbf{M} & \mathbf{\Sigma}_{b|a}^{-1}\end{array}\right)^{-1} = \left(\begin{array}{cc}\mathbf{\Sigma}_a & \mathbf{\Sigma}_a\mathbf{M}^\top \\ \mathbf{M}\mathbf{\Sigma}_a & \mathbf{\Sigma}_{b|a} + \mathbf{M}\mathbf{\Sigma}_a\mathbf{M}^\top\end{array}\right). \tag{B.25}$$

Proof. Being the vector

$$\mathbf{x} = \left(\begin{array}{c}\mathbf{x}_a \\ \mathbf{x}_b\end{array}\right) \tag{B.26}$$

and its joint distribution, from the Theorems 1 and 2,

$$p(\mathbf{x}) = p(\mathbf{x}_b|\mathbf{x}_a)p(\mathbf{x}_a) = \frac{(2\pi)^{-(n_a+n_b)/2}}{\sqrt{\det \mathbf{\Sigma}_{b|a} \det \mathbf{\Sigma}_a}} \exp\left\{-\frac{1}{2}E\right\} \tag{B.27}$$

where

$$E = (\mathbf{x}_b - \mathbf{M}\mathbf{x}_a - \mathbf{d})^\top \mathbf{\Sigma}_{b|a}^{-1} (\mathbf{x}_b - \mathbf{M}\mathbf{x}_a - \mathbf{d}) + (\mathbf{x}_a - \boldsymbol{\mu}_a)^\top \mathbf{\Sigma}_a^{-1} (\mathbf{x}_a - \boldsymbol{\mu}_a) . \tag{B.28}$$

Rewriting $\mathbf{x}_b - \mathbf{M}\mathbf{x}_a - \mathbf{d}$ as f and $\mathbf{x}_a - \boldsymbol{\mu}_a$ as e, we have

$$\begin{aligned} E &= (f - \mathbf{M}e)^\top \mathbf{\Sigma}_{b|a}^{-1} (f - \mathbf{M}e) + (e)^\top \mathbf{\Sigma}_a^{-1} (e) \\ &= e^\top \left(\mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} \mathbf{M} + \mathbf{\Sigma}_a^{-1} \right) e - e^\top \mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} f - f \mathbf{\Sigma}_{b|a}^{-1} \mathbf{M} e + f^\top \mathbf{\Sigma}_{b|a}^{-1} f \\ &= \begin{pmatrix} e \\ f \end{pmatrix}^\top \begin{pmatrix} \mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} \mathbf{M} + \mathbf{\Sigma}_a^{-1} & -\mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} \\ -\mathbf{\Sigma}_{b|a}^{-1} \mathbf{M} & \mathbf{\Sigma}_{b|a}^{-1} \end{pmatrix} \begin{pmatrix} e \\ f \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{x}_a - \boldsymbol{\mu}_a \\ \mathbf{x}_b - \mathbf{M}\boldsymbol{\mu}_a - \mathbf{d} \end{pmatrix}^\top \mathbf{R}^{-1} \begin{pmatrix} \mathbf{x}_a - \boldsymbol{\mu}_a \\ \mathbf{x}_b - \mathbf{M}\boldsymbol{\mu}_a - \mathbf{d} \end{pmatrix} \end{aligned} \tag{B.29}$$

By (B.11) we have that

$$\begin{aligned} \det \mathbf{R}^{-1} &= \det \left(\mathbf{\Sigma}_{b|a}^{-1} \right) \det \left(\mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} \mathbf{M} + \mathbf{\Sigma}_a^{-1} - \mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} \mathbf{\Sigma}_{b|a} \mathbf{\Sigma}_{b|a}^{-1} \mathbf{M} \right) \\ &= \det \left(\mathbf{\Sigma}_{b|a}^{-1} \right) \det \left(\mathbf{\Sigma}_a^{-1} \right) \\ &= \frac{1}{\det \left(\mathbf{\Sigma}_{b|a} \right) \det \left(\mathbf{\Sigma}_a \right)} \end{aligned} \tag{B.30}$$

Finally by inspection of (B.29), we take the mean of $\mathbf{x}_b$ as $\mathbf{M}\boldsymbol{\mu}_a + \mathbf{d}$ which concludes the proof. □

With the results of Theorems 1, 2 and 3 we get the following corollary

Corollary 1. *Being $\mathbf{x}_b$ conditioned on $\mathbf{x}_a$ and Gaussian distributed as*

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a | \boldsymbol{\mu}_a, \mathbf{\Sigma}_a), \quad p(\mathbf{x}_b | \mathbf{x}_a) = \mathcal{N}\left(\mathbf{x}_b | \mathbf{M}\mathbf{x}_a + \mathbf{d}, \mathbf{\Sigma}_{b|a} \right) \tag{B.31}$$

$\mathbf{M}$ a constant matrix and $\mathbf{d}$ a constant vector, both with the appropriate dimensions. Then conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ is given by

$$p(\mathbf{x}_a | \mathbf{x}_b) = \mathcal{N}\left(\mathbf{x}_a | \boldsymbol{\mu}_{a|b}, \mathbf{\Sigma}_{a|b} \right) \tag{B.32a}$$

with

$$\boldsymbol{\mu}_{a|b} = \mathbf{\Sigma}_{a|b} \left(\mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} (\mathbf{x}_b - \mathbf{d}) + \mathbf{\Sigma}_a^{-1} \boldsymbol{\mu}_a \right) \tag{B.32b}$$

$$\mathbf{\Sigma}_{a|b} = \left(\mathbf{\Sigma}_a^{-1} + \mathbf{M}^\top \mathbf{\Sigma}_{b|a}^{-1} \mathbf{M} \right)^{-1} . \tag{B.32c}$$

The marginal density of $\mathbf{x}_b$ is given by

$$p(\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_b | \boldsymbol{\mu}_b, \mathbf{\Sigma}_b) \tag{B.33a}$$

with

$$\boldsymbol{\mu}_b = \mathbf{M}\boldsymbol{\mu}_a + \mathbf{d}, \quad \mathbf{\Sigma}_b = \mathbf{\Sigma}_{b|a} + \mathbf{M}\mathbf{\Sigma}_a \mathbf{M}^\top . \tag{B.33b}$$

References

[Bishop, 2006] Bishop, C. M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

[Graybill, 1983] Graybill, F. (1983). Matrices with applications in statistics. Wadsworth statistics - probability series. Wadsworth International Group.

[Rasmussen and Williams, 2005] Rasmussen, C. E. and Williams, C. K. I. (2005). Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press.

[Schön and Lindsten, 2011] Schön, T. B. and Lindsten, F. (2011). Manipulating the multivariate gaussian density. Technical report, Division of Automatic Control, Linköping University, Sweden.