Bayesian Model Selection

by

Markov Chain Monte Carlo methods

applied to

Neural Networks and Linear models

Master Thesis
by
Thomas Fabricius

SECTION FOR DIGITAL SIGNAL PROCESSING
DEPARTMENT OF MATHEMATICAL MODELLING
TECHNICAL UNIVERSITY OF DENMARK

15. December 1999

Abstract

This Master Thesis examines the possibility to approximate Bayesian models and estimate Bayesian model probabilities. Further it discusses how Bayes performs compared to traditional methods. Various sampling methods to generate samples from distributions has been examined: Rejection, Metropolis, Gibbs and Hybrid Monte Carlo sampling. Different methods to estimate the model probabilities has been examined: Importance sampling, thermodynamic integration, bridge sampling and path sampling. Path sampling was chosen since it can be proven to cover all the other methods. Bayes and Maximum Likelihood was used in linear and feed forward regression networks. Maximum Likelihood methods suffers from over fitting, suggesting the use of Occams-razor to reduce model complexity, the Bayesian approach does not. Pure Maximum Likelihood model selection proves inconsistency with test error, Bayes does not when having closed form solutions but penalized Maximum Likelihood performs also very well. The path sampling approximation to the evidence can not be proven either to be consistent or inconsistent with the test error, because it is too variate and it depends too much on its starting temperature. This is due to sampling problems. An ensemble approximation for the feed forward network posterior has been derived, which seems promising, also in a model selection setup where it can be combined with Importance sampling.

Keywords: Generalization error, Kullback-Leibler, Normalizing constants, Bayes-factors, evidence, phase transitions, over fitting, Occams-razor, Markov chains, Rejection, Metropolis, Gibbs and Hybrid Monte Carlo sampling, thermodynamic integration, bridge sampling and path sampling, regression, artificial neural network, linear models, Bayes learning, Maximum Likelihood, decision and hypotheses test, ensemble approximation, variational approximation

Resumé

Denne kandidat afhandling undersøger mulighederne for at estimere Bayesianske modeller og model sandsynligheder. Endvidere diskuteres hvorledes Bayes yder sammenlignet med traditionelle metoder. Forskellige samplings metoder til at generere samples fra fordelinger er blevet undersøgt: Rejection, Metropolis, Gibbs og Hybrid Monte Carlo sampling. Forskellige metoder til at estimere model sandsynligheden er blevet undersøgt: Importance sampling, termodynamisk integration, bridge sampling and path sampling. Path sampling blev valgt da denne kan vises at dække alle de andre. Bayes og Maksimum Likelihood blev brugt i lineære og feed-forward regressions netværk. Maksimum Likelihood metoder lider af over fitning, foreslående brugen af Occams barberkniv, den Bayesianske fremgangsmåde gør ikke. Maksimum Likelihood model valg udviser inkonsistens med test fejl, Bayes gør ikke når løsningen findes pålukket form. Path sampling approksimationen kan ikke bevises at være konsistent eller inkonsistent med test fejlen, fordi variationen er stor og da den afhænger af start temperaturen. Dette er p.g.a. samplings problemer. En ensemble approksimation for feed-forward netværket er blevet udledt som ser lovende ud.

Nøgleord: Generalisering fejl, Kullback-Leibler, Normaliserings konstant, Bayes faktorer, Evidence, faseovergang, overfitting, Occams barberkniv, Markov Kæder, Rejection, Metropolis, Gibbs og Hybrid Monte Carlo Sampling, termodynamisk integration, Bridge sampling og Path Sampling, regression, kunstige neurale netværk, lineære modeller, Bayes indlæring, Maksimum Likelihood, Beslutning og hypotese test, ensemble approksimering, variations approksimering.

Preface

This master thesis serves as documentation for the final assignment in the requirements to achieve the degree *Master of Science*. The thesis has a work load of 35 ETC-points out of 300 ETC-points for the entire study.

The thesis is worked out at Section for Digital Signal Processing, Department of Mathematical Modelling, Technical University of Denmark. Responsible thesis supervisor Associated Professor Lars Kai Hansen.

Thesis overview

The thesis consist of seven chapters and three appendices. The first chapter act as an introduction to the field of Bayesian analysis and a description of the theses considered. The following two chapters describes shortly some concepts in statistical modelling and Hypotheses testing without going into sophisticated details. From the more basic chapters follows two chapters of more practical interest. One of the chapters is devoted to the theory of drawing samples from general distributions. The other is advocated the theory of estimating model evidences by using the sampling methods. The sixth chapter is the experimental chapter describing the use of linear and Artificial Neural Networks in regression problems. The various methods described in the two preceding chapters are used. The last chapter contains the conclusion. The first appendix contains the derivations of the linear analytic Bayesian model. The second appendix shows the normalization of the gamma posteriors at various temperatures. The last appendix shows the derivations of the Ensemble method for the Artificial Neural Network.

Acknowledgements

I want to thank the staff, former staff members, Ph.D.s and other students at Digital Signal Processing. In particular a special thank to Carl Edward Rasmussen for the introduction to and teaching in Bayesian concepts and sampling procedures, and for patience when my questions was down earth. Another to have a warm thank is Pedro H. Sørensen whom helped me through the starting phase to get over the starting problems when learning Bayesian theory. Of course a special thank to my supervisor Lars Kai Hansen for adding perspective into the thesis and guidance to move in the right directions. Thanks to Jan Larsen for discussions on Maximum Likelihood versus Bayes. Also Johannes K. Nielsen has been a major help in the physic concepts of sampling methods. Ulrik Kjems for his large patience when Linux did not obey my orders but his.

Furthermore a special thank to Claus Svarer for the possibility to join Brain99 and BrainPET99. Another thank to "Rektors Rejsedispostions Beløb", "Familien Hede Nielsens Fond" and "Digital Signal Processing" for economical aid making it possible to visit Minneapolis VA Medical Center and joining the 1999 Minnesota Workshops. A special thank to Stephen C. Strother at Minneapolis VA Medical Center.

Lyngby, 15. December 1999

Thomas Fabricius

Contents

1 Introduction **1**

2 Statistical Modelling **3**

2.1 Concepts of Statistical Modelling 3

2.1.1 What is statistical modelling ? 3

2.1.2 Choice of loss function 5

2.2 What is Bayesian Modelling? 5

2.2.1 Priors in Bayesian Modelling 6

2.2.2 When is Bayesian modelling generalization optimal ? . 7

2.3 Bayesian Modelling versus Maximum Likelihood modelling . . 7

2.3.1 Stylistic comparison 7

2.3.2 Temperated learning includes Bayes and Maximum Likelihood 8

2.3.3 Ways to solve the Bayesian Model Approach 12

3 Model Selection and Decision **15**

3.1 Model Selection to reduce over fitting 15

3.2 Model Selection and Decision in hypotheses test 16

4 Sampling Theory **19**

4.1 Specifying the problem 19

4.2 Importance sampling 20

4.3 Rejection sampling 21

4.4 Markov Chain methods 21

4.4.1 Gibbs sampling 23

4.4.2 Metropolis sampling 25

4.4.3 Hybrid Monte Carlo sampling 28

4.4.4 Example: 2D-Gaussian by Hybrid Monte Carlo 33

5 Calculating the Model Evidence: Theory **37**

5.1 Evidence by Importance Sampling 37

5.1.1 Approximating the normalizing constant to a 1D Gaussian 38

5.2 Evidence by Bridge Sampling methods 39

5.3 Path Sampling 40

5.3.1 Connection to Thermodynamic Integration 42

5.3.2 Connection to Bridge Sampling 42

5.4 Choosing the intermediate systems 42

5.5 Choice of sampling density 43

6 Regression models **45**

6.1 The regression model 45

6.1.1 The noise model 46

6.2 The Linear Model 47

6.2.1 Maximum likelihood for the linear model 48

6.2.2 Analytic Bayesian derivation 49

6.2.3 Monte Carlo Estimate 51

6.2.4 A toy problem: Selecting the number of inputs 53

6.2.5 Comparison of the three methods 64

6.3 Artificial Neural Network regression 65

6.3.1 The Network Architecture 65

6.3.2 The Maximum Likelihood solution 66

6.3.3 The Bayesian solution 67

6.3.4 Ensemble estimation 68

6.3.5 Monte Carlo estimation 72

6.3.6 Robot arm prediction and selection of neural network complexity 74

6.3.7 Comparison of the Maximum Likelihood approach and the Monte Carlo estimation 87

7 Conclusion **89**

A Analytic derivation for the linear model **93**

A.1 Predictive distribution: Linear model 93

B Conditional distributions of the linear model **97**

C Ensemble estimate of the ANN **99**

C.1 Approximating the ANN 99

C.2 Algorithm for ensemble learning 103

Bibliography **104**

List of Figures

2.1 Stylistic plot of two posteriors arising from different data set . 9
2.2 Bias-Variance tradeoff when learning the mean of a Gaussian . 12

4.1 Gibbs sampling from 2D-Gaussian 24
4.2 Metropolis sampling from a 2D-Gaussian 27
4.3 Hybrid Monte Carlo Sampling from a 2D-Gaussian 33
4.4 Leapfrog steps when sampling from a 2D-Gaussian 34
4.5 Leapfrog steps when sampling from a highly correlated 2D-Gaussian 35

5.1 Importance sampling estimates of normalizing constant 39

6.1 Linear model 47
6.2 Maximum Likelihood for the linear mode, likelihood, penalized likelihood, training variance and test variance 56
6.3 Logarithm of predictive distribution for linear model trained by Maximum Likelihood 57
6.4 Evidence by analytic Bayes solution to the linear model, together with the priors 58
6.5 Logarithm of predictive distribution in the analytic Bayes solution to the linear model 59
6.6 Linear regression input selection by MCMC 60
6.7 Linear model samples to estimate the evidence 61
6.8 Noise level and weights level as function of inverse temperature 62
6.9 Doint distribution of linear model hyperparameters 63
6.10 Robot arm output plotted against each other 75

6.11 Robot arm data plotted against the inputs 76
6.12 Choosing weight decay in ML ANN 77
6.13 ANN Training error in ML setup 78
6.14 ANN Test error estimate by FPE in ML setup 79
6.15 ANN Test error in ML setup 80
6.16 Convergence to stationary distribution 81
6.17 Test errors in Bayesian ANN 82
6.18 Prediction from network 83
6.19 Step size in Hybrid Monte Carlo for different temperatures. . . 84
6.20 Log evidence estimates for ANN 85
6.21 Hysteresis in evidence estimation 86
6.22 Outlier data point explanation 87

1. Introduction

Bayesian modelling can bee dated back to Laplace. The name Bayes comes in due to the extensive use of Bayes theorem in these kind of modelling scenarios. Bayesian models is models averaged over all possible parameters, rather than a model with optimal parameters like Maximum Likelihood. The nice about averaging is it capability of removing over fitting. Over fitting calls for Occams-razor to reduce the number of parameters that should be fitted. For that reason model selection is a major part of traditional modelling. For the Bayesian modelling approach this should not be the motivation. I will try to show that using Bayes in regression problems will not over fit data. Further more will I examine how Bayes performs when selecting between competing models. The selection result should be consistent with the error on independent test data.

I will try to answer two hypotheses with Bayesian and Maximum Likelihood modelling and decision: In linear regression we often have a lot of input and some outputs, we then want to decide how many inputs that are used to model the output ? The other hypotheses is: in a single layer feed forward ANN regression setup we often think that a specific number of hidden units are optimal. We then want to ask how many hidden units are optimal? the belief is that the Bayesian modelling procedure should in general cases suggest infinite many hidden units, this belief is motivated by the assumption, that in general cases real data does not come from a finite ANN, we can approximate these data arbitrary close by using more and more hidden units [3]. Whereas the Maximum Likelihood approach would select a finite model due to over fitting of too large models.

I will examine the possibility to estimate and use Bayesian model selection criteria. The Bayesian model selection approach is to calculate probabilities of the models of interest only given the training data or calculating model probability ratios called Bayes factors. Different methods to estimate the model probabilities will be examined: rejection sampling, thermodynamic integration, bridge sampling and path sampling. Both Bayesian modelling and Bayesian model selection often end up with multi dimensional integrals that can not lead to closed form expressions. For that reason methods like the evidence approximation [10], sampling methods [15], [13] and ensemble methods [11], [9] will bee used to approximate the posterior distribution. The results from the Bayesian model selection criteria are compared to the selection obtained using independent validation or test data on the individual Bayesian models; the last being a well accepted method known to yield unbiased decisions. The results from the Bayesian analysis is compared to their Maximum Likelihood counterpart. Since model selection leans on the modelling it self, a lot of the thesis is spend on this topic.

I will use The Hybrid Monte Carlo method together with Gibbs sampling [16], [15] to simulate data from the posterior of the ANN parameters. This method uses gradient information to move around in the posterior. The method is for that reason effective to suppress random walk behavior and hence effective in correlated distributions.

2. Statistical Modelling

In this chapter the concept of statistical modelling will be sketched. Some similarities for different inference methods will be outlined.

2.1 Concepts of Statistical Modelling

2.1.1 What is statistical modelling ?

Statistical models are used when the phenomena we try to model is of stochastic rather than of deterministic nature. The stochastic nature implies that statements have some uncertainty, i.e. it has some distribution. Statements can be assigned to stochastic variables. So a part of the modelling procedure is to decide how the stochastic variables are distributed. This can be carried out in many ways, sometimes an exact distribution can be derived for the problem other times only a family of distributions can be derived and some times there are no way to choose the distribution nor even its family, in this case one often chooses a model family by some criteria like a form that is believed to fit and/or it's ease of computation. If a stochastic variable interacts with other stochastic variables, we can try to model the dependencies between these stochastic variables. The dependencies and distributions can be described by some parameters, which can be believed to have some exact value or a distribution. If we believe the parameter is exact, it is not for sure that we can determine it's exact value, if we can, it is deterministic, if not it is stochastic just like the quantity that is stochastic by nature.

When starting modelling we assume having a training set containing a number of stochastic variables $\mathbf{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$ all coming from the same phenomena with distribution $p\left(\mathbf{x}, \mathbf{H}\right)$, where $\mathbf{H}$ is some model description. The different possible models will have some distribution $\pi\left(\mathbf{H}\right)$. Eventually $\pi\left(\mathbf{H}\right) = \delta\left(\mathbf{H} - \mathbf{H}_0\right)$ hence $p\left(\mathbf{x}\right) = p\left(\mathbf{x} \left| \mathbf{H}_0 \right.\right)$. $\mathbf{H}_0$ can be a model family, where each member is described by some parameters $\boldsymbol{\omega}$ with some kind of distribution $\pi\left(\boldsymbol{\omega} \left| \mathbf{H}_0 \right.\right)$. By Bayes rule we will have

$$p\left(\mathbf{x}\right) = p\left(\mathbf{x} \left| \mathbf{H}_0 \right.\right) = \int f\left(\mathbf{x} \left| \boldsymbol{\omega}, \mathbf{H}_0 \right.\right) \pi\left(\boldsymbol{\omega} \left| \mathbf{H}_0 \right.\right) d\boldsymbol{\omega} \tag{2.1}$$

If $p(\boldsymbol{\omega}\,|\mathbf{H}_0) = \delta(\boldsymbol{\omega}-\boldsymbol{\omega}_0)$ i.e. $\boldsymbol{\omega}$ has a specific value $\boldsymbol{\omega}_0$, then $p(\mathbf{x}) = p(\mathbf{x}\,|\mathbf{H}_0) = f(\mathbf{x}\,|\boldsymbol{\omega}_0, \mathbf{H}_0)$.

The whole modelling procedure is then the game of guessing a distribution based on the observed data $\mathbf{D}$, we could denote this guess by $\widehat{p}\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right)$, where $\widehat{\mathbf{H}}$ is the model we guess that describes our data. We could assign a loss $\mathcal{L}\left(\widehat{p}\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right), p(\mathbf{x})\right)$ to this guess, and calculate the expected loss for a specific training set

$$\Gamma\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right) = \int \mathcal{L}\left(\widehat{p}\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right), p(\mathbf{x})\right) p(\mathbf{x})\, d\mathbf{x} \tag{2.2}$$

Since we could face different training sets we could expect a loss of[1]

$$\Gamma\left(\mathbf{x}\,\middle|\widehat{\mathbf{H}}\right) = \iint \mathcal{L}\left(\widehat{p}\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right), p(\mathbf{x})\right) p(\mathbf{x})\, d\mathbf{x} p(\mathbf{D})\, d\mathbf{D} \tag{2.3}$$

This is sometimes referred as the expected generalization error.

If we know how to average distributions from different training sets, we could get a model $\langle\widehat{p}(\mathbf{x}\,|\mathbf{H})\rangle_{\mathbf{D}}$ i.e. averaged over all possible training sets. We could follow the idea of [8] and calculate the bias of the expected loss

$$\Gamma_B\left(\mathbf{x}\,\middle|\widehat{\mathbf{H}}\right) = \int \mathcal{L}\left(\left\langle\widehat{p}\left(\mathbf{x}\,\middle|\widehat{\mathbf{H}}\right)\right\rangle_{\mathbf{D}}, p(\mathbf{x})\right) p(\mathbf{x})\, d\mathbf{x} \tag{2.4}$$

we denote the loss by a B, since it tells us how biased our model in average would be. The variance of the expected loss can then be defined as

$$\Gamma_V\left(\mathbf{x}\,\middle|\widehat{\mathbf{H}}\right) \equiv \Gamma\left(\mathbf{x}\,\middle|\widehat{\mathbf{H}}\right) - \Gamma_B\left(\mathbf{x}\,\middle|\widehat{\mathbf{H}}\right) \tag{2.5}$$

The optimal $\widehat{p}\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right)$ is in general one that minimizes 2.3.

In the previous we just saw how we could make inferences about $\widehat{p}\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right)$, but in general we do not know $p(\mathbf{x})$ nor $p(\mathbf{D})$. So what we have to do is using a method to make inferences about $\widehat{p}\left(\mathbf{x}\,\middle|\mathbf{D},\widehat{\mathbf{H}}\right)$, that in general performs better than other methods, we say such a method performs uniform better than other methods. Unfortunately it is very unlikely that such a method exists. Instead we can talk about performance for a specific loss function, but even then it is seldom that a uniform better solution exists.

[1] If the training data are independent $p(\mathbf{D}) = \prod_{i=1}^{N} p(\mathbf{x}_i)$, but this is not necessarily true.

2.1.2 Choice of loss function

The choice of loss function is nearly as difficult as selecting the method to make inferences about $\widehat{p}\left(\mathbf{x}\left|\mathbf{D},\widehat{\mathbf{H}}\right.\right)$, and indeed these two topics interacts. The loss function should express some kind of prior information, like expressing that for some specific $\mathbf{x}$ intervals it would be very expensive if $\widehat{p}\left(\mathbf{x}\left|\mathbf{D},\widehat{\mathbf{H}}\right.\right)$ predict in this interval to often or to seldom. A common used cost function is the "log loss" which can be derived from the Kullback-Leibler distance

$$\mathcal{KL}\left(\widehat{p}\left(\mathbf{x}\left|\mathbf{D},\widehat{\mathbf{H}}\right.\right),p\left(\mathbf{x}\right)\right)=\int p\left(\mathbf{x}\right)\log\frac{p\left(\mathbf{x}\right)}{\widehat{p}\left(\mathbf{x}\right)}d\mathbf{x} \tag{2.6}$$

which splits into two terms, the "log loss" also called deviance

$$D\left(\widehat{p}\left(\mathbf{x}\left|\mathbf{D},\widehat{\mathbf{H}}\right.\right),p\left(\mathbf{x}\right)\right)=-\int p\left(\mathbf{x}\right)\log\widehat{p}\left(\mathbf{x}\right)d\mathbf{x} \tag{2.7}$$

and the residual called the entropy or "self entropy" of $p\left(\mathbf{x}\right)$

$$E\left(p\left(\mathbf{x}\right)\right)=\int p\left(\mathbf{x}\right)\log p\left(\mathbf{x}\right)d\mathbf{x} \tag{2.8}$$

The minimum of 2.3 with $\mathcal{KL}$ as loss function is located at the same $\widehat{p}\left(\mathbf{x}\left|\mathbf{D},\widehat{\mathbf{H}}\right.\right)$ as the minimum of the "log loss"

$$D\left(\mathbf{x}\left|\widehat{\mathbf{H}}\right.\right)=\iint-\log\widehat{p}\left(\mathbf{x}\left|\mathbf{D},\widehat{\mathbf{H}}\right.\right)p\left(\mathbf{x}\right)d\mathbf{x}p\left(\mathbf{D}\right)d\mathbf{D} \tag{2.9}$$

The loss is now relative to the self entropy. The "log loss" is mathematical convenient, since a lot of distributions are of exponential type. The "log loss" is often used as a measure for the relative model generalization, if an absolute measure is wanted then $\mathcal{KL}$ could be used as loss function.

2.2 What is Bayesian Modelling?

Bayesian methods provides a way to include a priori knowledge to a problem. Even when no a priori knowledge can be provided, Bayesian methods provides a kind of averaging. Averaging can be used to minimize 2.3 by introducing bias. Despite all these features, Bayesian methods are still not, like any other statistical modelling procedure, a single golden way of getting accurate results. The results are and will be of stochastic nature, just as any other procedure would yield.

Lets say, we have a set of data $\mathbf{D}$ coming from some distribution, and we want to predict a new example $\mathbf{x}$ also coming from this distribution. In a Bayesian framework, one always start out at the likelihood $f(\mathbf{D}|\,\boldsymbol{\omega},\mathbf{H})$ of the parameters $\boldsymbol{\omega}$ of the model $\mathbf{H}$ describing the data[2]. We want to make inferences about $\mathbf{x}$ based on $\boldsymbol{\omega}$. So if it is possible we want to calculate the distribution $k(\boldsymbol{\omega}|\,\mathbf{D},\mathbf{H})$. This can be carried out by the use of Bayes' rule

$$k(\boldsymbol{\omega}|\,\mathbf{D},\mathbf{H}) = \frac{f(\mathbf{D}|\,\boldsymbol{\omega},\mathbf{H})\,\pi(\boldsymbol{\omega}\,|\mathbf{H})}{\int f(\mathbf{D}|\,\boldsymbol{\omega}',\mathbf{H})\,\pi(\boldsymbol{\omega}'\,|\mathbf{H})\,d\boldsymbol{\omega}'} \tag{2.10}$$

$k(\boldsymbol{\omega}|\,\mathbf{D},\mathbf{H})$ is called the posterior of $\boldsymbol{\omega}$, $\pi(\boldsymbol{\omega}\,|\mathbf{H})$ is called the prior of $\boldsymbol{\omega}$. As seen, calculating $k(\boldsymbol{\omega}|\,\mathbf{D},\mathbf{H})$ is only possible when we can provide some a priori knowledge in the form $\pi(\boldsymbol{\omega}\,|\mathbf{H})$ about $\boldsymbol{\omega}$ before we ever saw any data. The prior should express some kind of believes on different $\boldsymbol{\omega}$, if we do not have any preferences we should set $\pi(\boldsymbol{\omega}\,|\mathbf{H})$ to be non-informative, which is one type of "Vague Bayes". In general this is not recommendable, since this is an improper prior, which means

$$z(\mathbf{H}) = \int \pi(\boldsymbol{\omega}'\,|\mathbf{H})\,d\boldsymbol{\omega}' \tag{2.11}$$

is divergent.

When $k(\boldsymbol{\omega}|\,\mathbf{D},\mathbf{H})$ is derived, the so called predictive distribution can be calculated

$$p(\mathbf{x}|\,\mathbf{H}) = \int f(\mathbf{x}|\,\boldsymbol{\omega}',\mathbf{H})\,k(\boldsymbol{\omega}'|\,\mathbf{D},\mathbf{H})\,d\boldsymbol{\omega}' \tag{2.12}$$

and used as a guess on the true distribution. What we can see is the Bayesian method have made the inference problem into a probabilistic inference problem. This method makes the prediction independent of the model parameters. In most modelling procedures this is wanted in others, where decisions are made upon the parameters, this is not. The decision problem, and how to solve it, is actual the major part of this thesis, and it will be faced again in chapter 5.

2.2.1 Priors in Bayesian Modelling

In Bayesian modelling priors are of major importance. The priors are provided to implement some kind of a priori knowledge. The strictly Bayesian method is to select the priors without knowledge to the actual data set. The

[2]The model does not necessarily have to be a single model, we could have different model candidates or said in another way a distribution $p(\mathbf{H})$ of models. If the latter is the case one could average over these.

major challenge is then to compile this information into a proper distribution. Many distributions can be compatible with the provided information, in this case the most convenient [3] one may be chosen. Often one can not provide any a priori knowledge i.e. meaning that one does not have any preferences towards any parameters. Two different approaches in selecting priors are conjugate priors and Jeffreys non-informative priors [19]. The conjugate priors are often preferable because they can lead to analytic tractable solutions, strictly this should not justify the use of them. The Jeffrey priors can be derived from the sampling distribution. The difficulties in selecting the right prior are often questioned by non-Bayesians.

2.2.2 When is Bayesian modelling generalization optimal ?

For any modelling procedure we must require that it is optimal in some sense. A derivation of when Bayes is optimal will here be given based on [7]. The average expected loss was defined by 2.3. When using the $\mathcal{KL}$-distances as loss function we know that this is minimized if

$$\widehat{p}\left(\mathbf{x}\left|\mathbf{D},\widehat{\mathbf{H}}\right.\right) = p\left(\mathbf{x}\right) \tag{2.13}$$

which of course is obvious.

Of course this is not very interesting, because we then would end up guessing on the teacher distribution. This can be some parameterized distribution, so we end up with a problem like the problem Maximum Likelihood tries to solve. The interesting in Bayesian modelling is of course: can we provide a teacher distribution that is "vague" compared to the true distribution either not by specifying the parameters but putting a prior on these that expresses our ignorance or by reformulating the model so it is more vague.

2.3 Bayesian Modelling versus Maximum Likelihood modelling

In this section Bayesian Model selection is compared to Maximum Likelihood selection. The comparison is performed by a stylistic example of the posterior and an example where we try to infer the mean of a 1D normal distribution with known variance. The derivation follows closely [7].

2.3.1 Stylistic comparison

A way to interpret the Maximum Likelihood in a Bayesian context [3], is when assuming the posterior $p\left(\boldsymbol{\omega}\middle|\,\mathbf{D}\right)$ is heavily peaked around the maximum

[3] which can be in a sense that makes it possible to find the final solution

Likelihood parameters $\boldsymbol{\omega}_{ML}$, then the predictive distribution becomes

$$p(x\,|\mathbf{D}) = \int p(x\,|\boldsymbol{\omega})\, p(\boldsymbol{\omega}\,|\mathbf{D})\, d\omega \tag{2.14}$$

$$\simeq \int p(x\,|\boldsymbol{\omega})\, \delta(\boldsymbol{\omega} - \boldsymbol{\omega}_{ML})\, d\omega \tag{2.15}$$

$$= p(x\,|\boldsymbol{\omega}_{ML}) \tag{2.16}$$

This approximation is only exact if we have an infinite amount of data, because then the posterior will converge towards the Dirac mass at the Maximum Likelihood parameters for that model.

When using non linear models like the ANN, it is not unusual that the posterior consists of several modes [3]. A stylistic example of such a posterior is shown on figure 2.1 [4]. On the figure is shown two posteriors arising from different data sets $\mathbf{D}_1$ and $\mathbf{D}_2$. When using Penalized Maximum Likelihood we would select the parameters marked $\boldsymbol{\omega}_{ML}\,|\mathbf{D}_1$ and $\boldsymbol{\omega}_{ML}\,|\mathbf{D}_1$. What we can see from the figure is, if we select the parameters that the first data set suggests, in a maximum likelihood sense, it would not be the best guess on the second data set. By using the Maximum Likelihood guess we model the posterior as the Dirac mass i.e. saying all other values of the parameter is a posterior very unlikely. This is a very strong belief on the suggested parameters. This belief arises even though most people know, that if they collect data a second time, they probably would get a different result. What the figure actually suggest is to make a continuum of models, and then weight them according to the posterior probability. This is actual what Bayesian inference does by the predictive distribution.

2.3.2 Temperated learning includes Bayes and Maximum Likelihood

The idea is to define a set of learning procedures characterized by a single parameter β, that both includes Maximum Likelihood and Bayes learning. In [7] this is performed by "temperating" the likelihood, such that the posterior for the model parameters becomes

$$p(\boldsymbol{\omega}\,|\beta, \mathbf{D}) = \frac{p^{\beta}(\mathbf{D}\,|\boldsymbol{\omega})}{\int p^{\beta}(\mathbf{D}\,|\boldsymbol{\omega}')\, d\boldsymbol{\omega}'} \tag{2.17}$$

where β is the inverse temperature. At $\beta = 1$ we have Bayes learning with uniform prior, at $\beta \to \infty$ we have maximum likelihood (ML) since this

[4]assuming a one parameter model

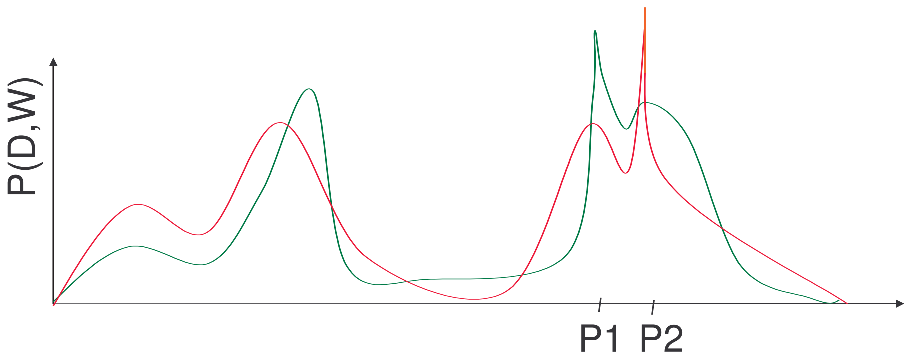


Figure 2.1: Stylistic plot of two posteriors arising from different data set

corresponds to cooling the system to absolute zero temperature. We could instead define a family of learning procedures for which the prior in the Bayes learning not necessarily have to be uniform

$$p(\boldsymbol{\omega}\,|\beta,\mathbf{D}) = \frac{p^{\beta}(\mathbf{D}\,|\boldsymbol{\omega})\,p^{1/\beta}(\boldsymbol{\omega})}{\int p^{\beta}(\mathbf{D}\,|\boldsymbol{\omega}')\,p^{1/\beta}(\boldsymbol{\omega}')\,d\boldsymbol{\omega}'} \tag{2.18}$$

Again at $\beta = 1$ we have Bayes learning, at $\beta \to \infty$ we have ML. Another family of learning procedures could be

$$p(\boldsymbol{\omega}\,|\beta,\mathbf{D}) = \frac{p^{\beta}(\mathbf{D}\,|\boldsymbol{\omega})\,p^{\beta}(\boldsymbol{\omega})}{\int p^{\beta}(\mathbf{D}\,|\boldsymbol{\omega}')\,p^{\beta}(\boldsymbol{\omega}')\,d\boldsymbol{\omega}'} \tag{2.19}$$

Which at $\beta = 1$ is Bayes learning and at $\beta \to \infty$ is Maximum A Posterior (MAP).

The predictive distribution is the distribution of interest

$$p(\mathbf{x}\,|\mathbf{D},\beta) = \int p(\mathbf{x}\,|\boldsymbol{\omega}')\,p(\boldsymbol{\omega}'\,|\beta,\mathbf{D})\,d\boldsymbol{\omega}' \tag{2.20}$$

We could now measure the generalizing ability of the predictive distribution by using 2.6 as loss measure by setting $\widehat{p}(\mathbf{x}) = p(\mathbf{x}\,|\mathbf{D},\beta)$

$$\mathcal{KL}(\widehat{p}(\mathbf{x}), p(\mathbf{x})) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\widehat{p}(\mathbf{x})} d\mathbf{x} \tag{2.21}$$

In learning theory $p(\mathbf{x})$ is sometimes referred to as the teacher distribution and $\widehat{p}(\mathbf{x})$ the student. The expected generalization error 2.3 is then

$$\Gamma = \iint \mathcal{KL}(\widehat{p}(\mathbf{x}), p(\mathbf{x}))\,p(\mathbf{x})\,d\mathbf{x}p(\mathbf{D})\,d\mathbf{D} \tag{2.22}$$

which can be split into 2.4 and 2.5

1D Normal distribution with known variance and Jeffrey prior

In this special problem we want to make inferences about a location parameter μ. In this case the teacher distribution is

$$p(x) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_0)^2\right) \tag{2.23}$$

The well known Jeffrey prior for a location parameter is

$$p(\mu) = k \tag{2.24}$$

In this case the learning procedures 2.17,2.18 and 2.19 are the same. The likelihood is then

$$p(\mathbf{D}\,|\mu) = \left(\frac{1}{2\pi\sigma^2}\right)^{N/2} \exp\left(-\frac{N}{2\sigma^2}(\overline{x}-\mu)^2\right) \tag{2.25}$$

with $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$. The posterior for a specific β is then

$$p(\mu\,|\mathbf{D},\beta) = \left(\frac{\beta N}{2\pi\sigma^2}\right)^{1/2} \exp\left(-\frac{\beta N}{2\sigma^2}(\overline{x}-\mu)^2\right) \tag{2.26}$$

The predictive distribution is then found by integrating out μ

$$\begin{aligned} p(x\,|\mathbf{D},\beta) &= \int p(x\,|\mu')\,p(\mu'\,|\mathbf{D},\beta)\,d\mu' \\ &= \left(\frac{1}{2\pi\sigma_\beta^2}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma_\beta^2}(\overline{x}-x)^2\right) \end{aligned}$$

with $\sigma_\beta^2 \equiv \sigma^2\left(1+(\beta N)^{-1}\right)$, so $\sigma_{ML}^2 = \sigma_\infty^2 = \sigma^2$ is always smaller than any of the other learning procedures with finite β, which is by the fact that the other procedures takes the uncertainty on μ into account by averaging. The $\mathcal{KL}$-distance between the teacher and the predictive distribution, which is also the generalization error, is then

$$\begin{aligned} \Gamma(\mathbf{D},\beta) &= \mathcal{KL}\left(p(x\,|\mathbf{D},\beta), p(\mathbf{x})\right) = \int p(\mathbf{x}) \log\frac{p(\mathbf{x})}{p(x\,|\mathbf{D},\beta)} d\mathbf{x} \\ &= \log(\frac{\sigma_\beta}{\sigma}) + \frac{1}{2\sigma_\beta^2}\left((\overline{x}-\mu_0)^2+\sigma^2\right) - \frac{1}{2} \end{aligned}$$

The expected generalization from facing different training set is then found by averaging with respect to the sampling distribution $\overline{x} \sim \mathcal{N}\left(\mu_0, \sigma^2/N\right)$

$$\Gamma\left(\beta\right) = \int \Gamma\left(\mathbf{D},\beta\right) p\left(\mathbf{D}\left|\mu_0\right.\right) d\mathbf{D} \tag{2.27}$$

$$= \log\left(\frac{\sigma_\beta}{\sigma}\right) + \frac{\sigma^2}{2\sigma_\beta^2}\left(\frac{1}{N}+1\right) - \frac{1}{2} \tag{2.28}$$

The learning procedure that performs best is the one that minimizes 2.27. So finding $\frac{\partial\Gamma(\beta)}{\partial\beta} = 0$ yields

$$\beta = 1 \tag{2.29}$$

i.e. Bayes learning. As seen this happens independently of μ_0, which means no matter which exact teacher that creates the data then Bayes is generalization optimal, that is even if there exists a distribution of teachers. This only happens in very few setups. So in the case of location parameters Bayes learning performs uniformly better than any other temperated method including maximum likelihood. The bias of 2.27 can now be calculated [8] by first finding the average distribution over different data set

$$\overline{p}\left(x\right) = Z^{-1} \exp\left(\int p\left(\mathbf{D}\left|\mu_0\right.\right) \log p\left(x\left|\mathbf{D}\right.\right) d\mathbf{D}\right) \tag{2.30}$$

By using the sampling distribution we get

$$\overline{p}\left(x\left|\mu_0\right.\right) = \frac{1}{\sqrt{2\pi\sigma_\beta^2}} \exp\left(-\frac{1}{2\sigma_\beta^2}\left(x-\mu_0\right)^2\right) \tag{2.31}$$

The average $\mathcal{KL}$-error of this distribution for different draws of x from $p\left(x\left|\mu_0\right.\right)$ is then

$$\Gamma_B\left(\beta\right) = -\int p\left(x\left|\mu_0\right.\right) \log \frac{p\left(x\left|\mu_0\right.\right)}{\overline{p}\left(x\left|\mu_0\right.\right)} dx \tag{2.32}$$

which equals

$$\Gamma_B\left(\beta\right) = \log\frac{\sigma}{\sigma_\beta} + \frac{\sigma^2}{2\sigma_\beta^2} - \frac{1}{2} \tag{2.33}$$

And the variance can now be calculated as $\Gamma_V\left(\beta\right) = \Gamma\left(\beta\right) - \Gamma_B\left(\beta\right)$

$$\Gamma_V\left(\beta\right) = \frac{\sigma^2}{2N\sigma_\beta^2} \tag{2.34}$$

The generalization error and its bias-variance contributions is shown on figure 2.2. As we know introducing bias i.e. regularization the model can reduce

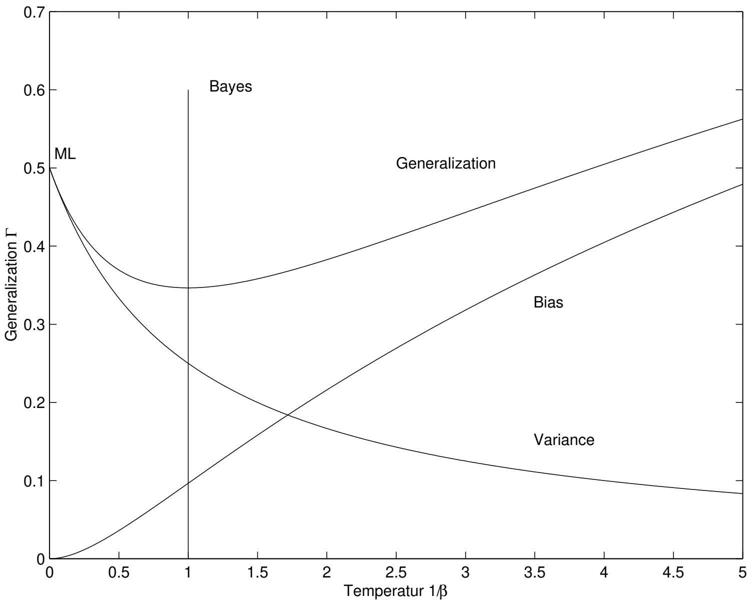


Figure 2.2: Tradeoff between bias and variance as function of temperature. At zero temperature we have the Maximum Likelihood solution and at temperature one we have the Bayes solution. The figure is made based on a single example. When using more examples the difference between the learning algorithms becomes smaller, but Bayes solution will always have the lowest Generalization error

the average error. The figure shows the bias-variance trade-off as function of temperature. What we see is Bayes produces the lowest generalization error. The explanation, that Bayes performs better than ML in this setup, is because Bayes introduces a certain amount of bias to reduce the variance.

2.3.3 Ways to solve the Bayesian Model Approach

The Bayesian modeling procedure is as described to put priors on unknown parameters instead of finding some optimal parameters in the sense that they maximizes the likelihood. What we see is the problem reduces to solving a often high dimensional integral over the model parameters. These integrals can, if not solved analytic, be solved by sampling see chapter 4 or another numerical integration method. Often the sampling method is superior in high dimensions. The nice thing about sampling is it can be proven to produce asymptotically correct answers as the number of samples produced grows. Other methods that can be used is f.x. the evidence approximation [10] where one finds a mode or several modes [5] for some particular parameters given the other parameters and approximate those by a Gaussians. When the approximation is found then the other parameters are reestimated to maximize what is called the evidence. The evidence is the likelihood integrated with respect to the posterior of the parameters we did make a Gaussian approximation for, this can lead to a closed form expression for the evidence. When these parameters are reestimated the idea is to find the new modes approximate them, reestimate the other parameters e.t.c. Lately [11] the evidence approximation to Artificial Neural Networks is proven to be much the same as what is called the ensemble approximation. The ensemble method is relative new in the Bayesian community. It is also called variational approximation. Ensemble theory is in physics an established method. In appendix C have I derived an ensemble approximation to the Artificial Neural Net which have the potential compared to the evidence approximation to catch some more variations. It builds on a method which approximate the true posterior by one having a simpler form by minimizing the Gibbs free energy, which is the same as the KL-distance between the true posterior and the approximation. The simpler form is Gaussian in the weights and the reestimation of the other parameters is the mean of their respective posteriors. This lead to an iterative scheme just like the evidence approximation, but no maximization has been performed.

[5] This can be carried out by finding the minima by some favorite method

3. Model Selection and Decision

In this chapter some of the differences between the classical and Bayesian decision methods will be explained. The maximum likelihood approach will be compared to its counterpart in Bayesian statistics the Bayes factor. In all kinds of decision there is a cost or opportunity loss connected to a single decision, this will only be addressed shortly. Of course we always want our decision strategy to perform uniformly better than any other strategy. But usually this is not possible, so then we have to choose between the possible strategies by choosing the one that minimizes the maximal possible error or minimizes the mean error e.t.c.

There are two main purposes for Model Selection and Decision: In maximum likelihood modelling we want to reduce the number of parameters to prevent over fitting i.e. fitting noise by some of the free parameters, so we have to select the model complexity. The other purpose is when we have a set of competing hypotheses and we from data want to select amongst these.

3.1 Model Selection to reduce over fitting

This problem is of major interest when using large flexible models as Artificial Neural Networks (ANN), Gaussian Mixture Models e.t.c. where our model hypotheses are very vague. We know Maximum Likelihood has some fine asymptotic properties. The Bayesian method where averaging with respect to the posterior yields the predictive distribution, has the nearly same properties, since the priors will be overwhelmed by data and tend towards a delta function on the true parameters if there exists such true parameters. As discussed in the previous chapter, even if the model is correct and we only have to infer or select the parameters from a finite data set, we can not justify using the Maximum Likelihood parameters since it has probability zero to guess the real parameter. When the model is not the correct model, but it have properties like the ANN which in the limit can approximate any function arbitrary close [3] by using more and more hidden units, it should be possible to approach the limit and reach the ”right” model. In general inferring the parameters in such large models is an ill posed problem. The Maximum Likelihood solution will begin to fit the noise and the Bayes solution will fall back on its priors. If the priors are selected in a reasonable way it should be possible to add in as many units as we have computation time to manage and still gain performance. But since we can not calculate the predictive distribution arbitrarily close, due to finite precision, we must expect that the performance flattens out for more complex models.

The solution to prevent over fitting in the Maximum Likelihood setting is to penalize the model, which act much as adding a prior in the model. But even if the penalty is the exact same as the prior in the Bayesian setting, Maximum Likelihood will still have the problem of selecting the "true" parameters which have probability zero even if there exists such parameters. Another way to penalize complex models is to use some information criterion like Akaike Information Criterion (AIC) [3], Bayes Information Criterion (BIC) and Generalized Prediction Error (GPE) [3]. But all these methods will still play the role of a kind of prior since they regularize the model. But again, even if the penalty was the "true" one, Maximum Likelihood will suffer of over fitting by guessing on a point.

3.2 Model Selection and Decision in hypotheses test

The other problem in model selection and decision is the hypotheses test. To discuss this we need some definitions. When talking about hypotheses test, we have two hypotheses the null hypothesis often denoted H_0 and the alternative often denoted H_1. The hypotheses can either be simple, combined or integrated. If the hypotheses is simple, it is characterized by a point. A simple hypotheses could be: "Is there exact 3 input to this regression model ?". The alternative can then either be simple, combined or integrated. A simple alternative would then be: "Or is there exact 4 ?". The combined alternative could be: "Or is there exact 0, 1, 2 or 4 e.t.c. ?" i.e. many alternatives. The integrated alternative would be: "Or is there exact 0 and 1 and 2 and 3 and 4 and e.t.c.", the integrated alternative could be understood as sometimes we see three inputs other times five e.t.c. If the null hypothesis is rejected, we are not interested in knowing which of the possible alternatives are the right one. Or if we are it then becomes the null hypothesis.

What can the result from doing such a test be ? there are four possible outcomes: We can accept a null hypothesis that is true, reject a null hypothesis that is true error type I, accept a null hypothesis that is false error type II, and at last we can reject a null hypotheses that is false. We often talk about the probability of the different events i.e. the probability of accepting the null hypothesis if it is true e.t.c. We will nearly always have to accept that, if the probability of accepting a null hypothesis that is true should be high, then the probability of accepting the null hypothesis if it is not true will also be high. Since these four probabilities often depends on the specific hypothesis f.x. it can be very difficult to calculate the probability of rejecting the null hypothesis if the alternative is true [1], since there are many alternatives if it is combined. For that reason the power of the test is defined as: "Probability of rejecting the null hypotheses if a specific alternative is true", this means the power is a function of the different alternatives.

There are two traditional approaches in test theory. The first approach [19] builds on the Neyman-Pearson lemma. It says that there exists a uniform most powerful tests when, there exists a total ordering of the hypotheses in the null hypothesis and the alternative and we fix the maximum probability of type I error at α. What this means is, by all the possible null hypotheses one [2] hypothesis will have the largest tendency to produce samples that will be interpreted as arising from the alternatives, if we fix this to happen for a fraction α of samples that is really drawn from this null hypothesis, then we can construct a test that has lower power at all alternatives than any other test procedure. This is also true even if there exists a test which is specialized in rejecting the null hypothesis for a specific alternative being true. If this specialized test should be better to reject when a alternative is true, it must suffer from a higher rejection rate of the true null hypotheses.

Often there does not exist such a total ordering, then the second approach likelihood-ratios is used to produce a test. These tests builds on selecting the hypothesis amongst the null hypotheses and the alternative hypotheses that yields the highest likelihood.

This approach is much the same Bayesians use. In this case it is the maximum marginalized posterior that is used as selection criteria. The marginalization is with respect to the parameters out of interest. So we end up with a probability of each hypothesis. Of course all the same type of errors exists in the Bayesian framework. It is obvious that the Bayesian approach is the right method when either the null hypothesis or alternative is of what I call the integrated type.

The right way to hypotheses test is to assign a cost or opportunity loss to each of the different errors. Then we can apply a strategy: either making the decision that reduces the maximal cost called the min-max strategy or minimizing the average cost or another strategy. The problem is then to use a method that can find the solution to this strategy. How Bayes and

[1] the same as rejecting the null hypothesis if it is false

[2] since there exists a total ordering

Maximum Likelihood performs will fall back on the problem of estimating the model. Of course a reasonable decision can only be made if the model, which the decision is based upon, is estimated correct.

For a comparison of Bayesian and Maximum Likelihood tests see [19]. One major conclusion is the criticism of the standard frequentist tests, which often are used with out specifying actual costs which leads to the rhetorical use of 5% levels with out further e.d.o.

4. Sampling Theory

This chapter describes various methods used to generate samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$ from $p\left(\boldsymbol{\omega}\right)$. These samples can be used to solve high dimensional integration problems by a Monte Carlo estimate. As seen in chapter 6.3.5 this is one way to estimate the predictive distribution arising in Bayesian models. Another problem which can be solved by sampling is the model selection estimation chapter 5.

4.1 Specifying the problem

The problem to solve/estimate is

$$\Phi = E_{\boldsymbol{\omega}}\left[f\left(\boldsymbol{\omega}\right)\right] = \int f\left(\boldsymbol{\omega}\right) p\left(\boldsymbol{\omega}\right) d\boldsymbol{\omega} \tag{4.1}$$

i.e. the expectation of a function $f\left(\boldsymbol{\omega}\right)$ with respect to the distribution $p\left(\boldsymbol{\omega}\right)$. The integral is possible of high dimension. Of course it is possible to calculate the integral by normal numerical integration, but in general this is very hard in high dimensions.

Another way to solve the problem is to generate samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$ and then calculate a Monte Carlo estimate

$$\widehat{\Phi} = \frac{1}{N_s}\sum_{i=1}^{N_s} f\left(\boldsymbol{\omega}_i\right) \tag{4.2}$$

Some times we just want the samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$.

For some general distributions like the Gaussian, t-distribution, Gamma and exponential there exists some pseudo random number generators, that produces nearly independent samples [19]. They all build on the possibility to produce independent numbers from the uniform distribution on the interval $[0;1]$.

The problem to get samples from $p\left(\boldsymbol{\omega}\right)$ is challenging. This can sound peculiar since we have the canonical form of $p\left(\boldsymbol{\omega}\right)$ and hence can evaluate it in an arbitrary point. The problem can be even worse if we only know $p(\boldsymbol{\omega})$ up to a multiplicative constant

$$p(\boldsymbol{\omega}) = \frac{1}{c} q(\boldsymbol{\omega}) \tag{4.3}$$

where $c = \int q(\boldsymbol{\omega}')\, d\boldsymbol{\omega}'$.

In the next couple of sections some general methods to produce samples from arbitrary distributions $p(\boldsymbol{\omega})$ will be presented. Most of these builds on the uniform generator.

4.2 Importance sampling

The importance method can not be used to produce the samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$ from $p_s(\boldsymbol{\omega})$ but can be used to get an estimate $\widehat{\Phi}$ of 4.1. The method builds on the fact that it is possible to evaluate $p(\boldsymbol{\omega})$ up to a multiplicative constant i.e. evaluate $q(\boldsymbol{\omega})$. By having another distribution $p_s(\boldsymbol{\omega})$ from which we are able to generate samples and also can evaluate up to a multiplicative constant i.e. evaluate $q_s(\boldsymbol{\omega})$, it is possible to get the estimate. $p_s(\boldsymbol{\omega})$ is called the sampling distribution.

The simple idea is know to generate samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$ from $p_s(\boldsymbol{\omega})$. If these samples came from $p(\boldsymbol{\omega})$ i.e. $p(\boldsymbol{\omega}) = p_s(\boldsymbol{\omega})$, we could get the estimate by 4.2. But since the sampling density and the density of interest is not equal, the sampling density will have a higher density in some regions and hence a lower density in others than the density of interest. This fact give rise to an over representation and under representation of samples in these regions. But we know this over/under representation since we can evaluate the two distribution up to a multiplicative constant, the over/under representation can be calculated as the ratio

$$w_i = \frac{q(\boldsymbol{\omega}_i)}{q_s(\boldsymbol{\omega}_i)} \tag{4.4}$$

So know it is possible to weight each function evaluation by the above ratio and normalize

$$\widehat{\Phi} = \frac{\sum_{i=1}^{N_s} w_i f(\boldsymbol{\omega}_i)}{\sum_{i=1}^{N_s} w_i} \tag{4.5}$$

it can be proved that if $p_s(\boldsymbol{\omega})$ have non vanishing density in regions where $p_s(\boldsymbol{\omega})$ have density, 4.5 converges to the true mean. In chapter 5.1 an example of importance sampling will be used to estimate the normalizing constant of a Gaussian distribution.

4.3 Rejection sampling

Rejection sampling produces independent samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$. Again we have the distribution of interest $p\left(\boldsymbol{\omega}\right)$ that can be evaluated up to a multiplicative constant $q\left(\boldsymbol{\omega}\right)$. Rejection sampling builds on having a second distribution the proposal density $p_p\left(\boldsymbol{\omega}\right)$ that can be evaluated up to a multiplicative constant $q_p\left(\boldsymbol{\omega}\right)$ and from which we can generate samples. The proposal distribution should have the property that there exists a known constant k such that

$$kq_p\left(\boldsymbol{\omega}\right) > q\left(\boldsymbol{\omega}\right) \tag{4.6}$$

then it is possible to generate samples from $p\left(\boldsymbol{\omega}\right)$. It is now possible to decide whether a sample generated from the proposal density is representative or not and hence accept or reject it. To decide whether a proposed sample $\boldsymbol{\omega}^{(p)}$ should be accepted, we generate a number u from the uniform distribution on the interval $\left[0; kq_p\left(\boldsymbol{\omega}^{(p)}\right)\right]$. If u is larger than $q\left(\boldsymbol{\omega}\right)$ the sample is rejected otherwise it is accepted.

The method is useful if the proposal distribution is a good approximation to the distribution of interest, if not a lot of samples will be rejected. In high dimensions it is often very difficult to fulfill this requirement.

4.4 Markov Chain methods

The Markov Chain methods are used to iteratively generate samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$. The samples are not in general independent.

The method generates samples which in the Markov Chain context is called *states*. The generation of samples is iterative and hence we can order the samples by it's iteration index. We always start out at a sample/state $\boldsymbol{\omega}^{(0)}$ that eventually is drawn from some initial distribution. The idea is know to go to a new state $\boldsymbol{\omega}^{(n+1)}$ with density $p_t\left(\boldsymbol{\omega}^{(n+1)}\left|\boldsymbol{\omega}^{(n)}\right.\right)$ we call $p_t\left(\boldsymbol{\omega}^{(n+1)}\left|\boldsymbol{\omega}^{(n)}\right.\right)$ the *transition kernel*. That the new state only depends on the previous, is what characterize the setup as a Markov Chain. The transition probability can be independent of the iteration number i.e. if $\boldsymbol{\omega}^{(n)} = \boldsymbol{\omega}^{(k)}$ and $\boldsymbol{\omega}^{(n+1)} = \boldsymbol{\omega}^{(k+1)}$ then if and only if for all n, k, $\boldsymbol{\omega}^{(k)}$ and $\boldsymbol{\omega}^{(k+1)}$

$$p_t\left(\boldsymbol{\omega}^{(n+1)}\left|\boldsymbol{\omega}^{(n)}\right.\right) = p_t\left(\boldsymbol{\omega}^{(k+1)}\left|\boldsymbol{\omega}^{(k)}\right.\right) \tag{4.7}$$

the Markov Chain is said to be *homogeneous*.

The state at step $n+1$ can be interpreted as a sample from a distribution $p^{(n+1)}\left(\boldsymbol{\omega}\right)$ which over many Markov Chains can be calculated as

$$p^{(n+1)}\left(\boldsymbol{\omega}\right) = \int p^{(n)}\left(\boldsymbol{\omega}'\right) p_t^{(n)}\left(\boldsymbol{\omega}\left|\boldsymbol{\omega}'\right.\right) d\boldsymbol{\omega}' \tag{4.8}$$

when running a single Markov chain the distribution at iteration $n+1$ becomes

$$p^{(n+1)}(\boldsymbol{\omega}) = p^{(n)}\left(\boldsymbol{\omega}^{(n)}\right) p_t^{(n)}\left(\boldsymbol{\omega}\left|\boldsymbol{\omega}^{(n)}\right.\right) \tag{4.9}$$

so a single Markov chain started from the initial distribution $p^{(0)}(\boldsymbol{\omega})$ will in $n+1$ iteration reach the distribution

$$p^{(n+1)}(\boldsymbol{\omega}) = \prod_{k=0}^{n} p^{(k)}\left(\boldsymbol{\omega}^{(k)}\right) p_t^{(k)}\left(\boldsymbol{\omega}\left|\boldsymbol{\omega}^{(k)}\right.\right) \tag{4.10}$$

As the number of iterations increases the Markov Chain will eventually converge towards the distribution of interest what in Markov Chain terminology is called the *stationary* distribution or *invariant* distribution i.e. all further states from the stationary distribution will be samples from the distribution of interest i.e. invariant. We say that if and only if for all k

$$p^{(N)}(\boldsymbol{\omega}) = p^{(N+k)}(\boldsymbol{\omega}) \tag{4.11}$$

from a step N the actual distribution at that step and at further steps is invariant.

The Markov chain has an invariant distribution if we can show that the transitions fulfill *detailed balance* for the stationary distribution. Detailed balance means the probability to go from one state to another state is the same as going from the second state to the first i.e.

$$p^{(n)}(\boldsymbol{\omega}') p_t^{(n)}(\boldsymbol{\omega}\left|\boldsymbol{\omega}'\right.) = p^{(n)}(\boldsymbol{\omega}) p_t^{(n)}(\boldsymbol{\omega}'\left|\boldsymbol{\omega}\right.) \tag{4.12}$$

If this hold from a certain iteration N we can show invariance from this iteration since

$$\begin{aligned} p^{(N+1)}(\boldsymbol{\omega}) &= \int p^{(N)}(\boldsymbol{\omega}') p_t^{(N)}(\boldsymbol{\omega}\left|\boldsymbol{\omega}'\right.) d\boldsymbol{\omega}' && (4.13)\\ &= \int p^{(N)}(\boldsymbol{\omega}) p_t^{(N)}(\boldsymbol{\omega}'\left|\boldsymbol{\omega}\right.) d\boldsymbol{\omega}' && (4.14)\\ &= p^{(N)}(\boldsymbol{\omega}) \int p_t^{(N)}(\boldsymbol{\omega}'\left|\boldsymbol{\omega}\right.) d\boldsymbol{\omega}' && (4.15)\\ &= p^{(N)}(\boldsymbol{\omega}) && (4.16) \end{aligned}$$

A Markov chain that is homogeneous and fulfills detailed balance is said to be *reversible* if we could reverse all the previous steps and then end up in the start state.

As we see from above it is not enough to require invariance because how can we be sure that we reach the invariant distribution from an initial arbitrary distribution $p^{(0)}(\boldsymbol{\omega})$ in N iterations, even if we allow $N \rightarrow \infty$ we can not be sure this happens. If the Markov chain reaches the invariant distribution as $N \rightarrow \infty$ we say the Markov chain is *ergodic*. An ergodic Markov chain has only one invariant distribution which is called the *equilibrium* distribution. To proof ergodicity is not simple. But in general if the density $p^{(k)}\left(\boldsymbol{\omega}^{(k)}\right) > 0$ for all k and all $\boldsymbol{\omega}^{(k)}$ the chain is ergodic. Another property that will assure ergodicity is if $p_t^{(k)}\left(\boldsymbol{\omega}\left|\boldsymbol{\omega}^{(k)}\right.\right) > 0$ for some k we say the chain is *irreducible*. A more restrictive condition to assure ergodicity is if $p_t^{(k)}\left(\boldsymbol{\omega}\left|\boldsymbol{\omega}^{(k)}\right.\right) > 0$ for any k we say the Markov chain is *regular*. As we see regularity implies the Markov chain to be irreducible.

4.4.1 Gibbs sampling

The Gibbs sampling method is used to iteratively produce samples/states from the joint probability

$$p(\boldsymbol{\omega}) = \prod_{i=1}^{N_\omega} p\left(\omega_i \left| \{\omega_j\}_{j \neq i}\right.\right) \tag{4.17}$$

The samples are independent if we can produce independent samples from the conditional probabilities. Each state consists of N_ω transitions. This is done by at iteration $n+1$ to draw

$$\begin{align}
\omega_1^{(n+1)} &\sim p\left(\omega_1 \left| \left\{\omega_j^{(n)}\right\}_{j=2}^{N_\omega}\right.\right) \tag{4.18}\\
&\vdots \tag{4.19}\\
\omega_i^{(n+1)} &\sim p\left(\omega_i \left| \left\{\omega_j^{(n+1)}\right\}_{j=1}^{i-1}, \left\{\omega_j^{(n)}\right\}_{j=i+1}^{N_\omega}\right.\right) \tag{4.20}\\
&\vdots \tag{4.21}\\
\omega_{N_\omega}^{(n+1)} &\sim p\left(\omega_{N_\omega} \left| \left\{\omega_j^{(n+1)}\right\}_{j=1}^{N_\omega - 1}\right.\right) \tag{4.22}
\end{align}$$

We see to produce a new state for the Markov chain we make N_ω *Gibbs updates*.

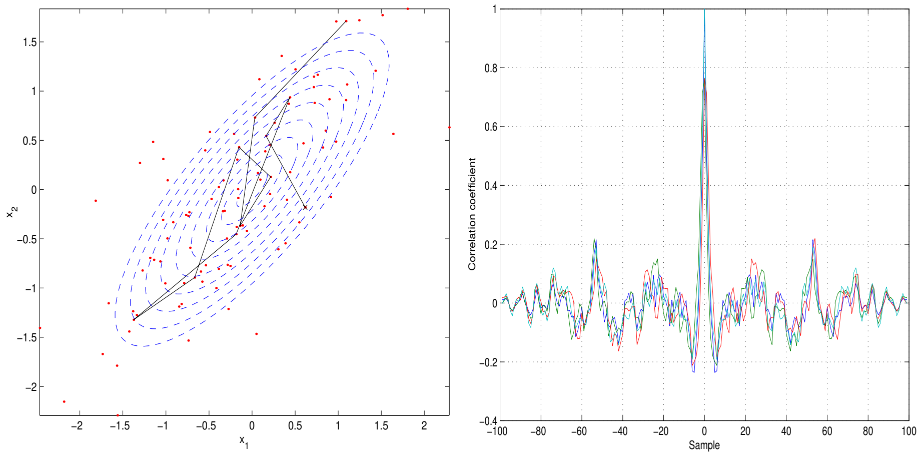
To show invariance we can show that the i^{th} Gibbs update leaves the desired distribution invariant. This is of course true for $\omega_k^{(n+1)}$ for $k \neq i$ since these are not changed at all in the i^{th} Gibbs update. And since

$\omega_i^{(n+1)} \sim p\left(\omega_i \,\middle|\, \left\{\omega_j^{(n+1)}\right\}_{j=1}^{i-1}, \left\{\omega_j^{(n)}\right\}_{j=i+1}^{N_\omega}\right)$ is producing a sample from the desired conditional distribution, the Markov chain has an invariant distribution. This can of course also be shown by showing that detailed balance holds. If the probability of getting a specific ω_i by a Gibbs update in some iteration is non zero for all possible draws for different i the Markov chain is irreducible and hence ergodic. If the conditional distribution is non zero for all ω_i and at all iterations the Markov chain is regular and hence ergodic.

Example: correlated 2D-Gaussian

The Gibbs method is used to sample from a 2D-Gaussian with zero mean and covariance

$$\begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} \tag{4.23}$$



(a) (b)

Figure 4.1: a: Gibbs sampling from a 2D Gaussian with variance 1 in both directions and correlation $\rho = 0.8$. The contours are plotted with equal logarithmic density difference. Every state is taken as a sample. The 10 first samples are connected by the trajectory. b: correlation between sample. As seen there is still some correlation between each sample. The three plots are correlation from each direction and their cross correlation.

To make the Gibbs updates we have to calculate the conditional distributions.

$$p(x_1|x_2) \propto \exp -\frac{1}{2} \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \tau_{11} & \tau_{12} \\ \tau_{12} & \tau_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{4.24}$$

$$\propto \exp -\frac{1}{2} (\tau_{11}x_1 + 2x_1x_2\tau_{12}) \tag{4.25}$$

$$\propto \exp -\frac{\tau_{11}}{2} \left(x_1 - \left(-x_2 \frac{\tau_{12}}{\tau_{11}} \right) \right)^2 \tag{4.26}$$

i.e. $x_1 \sim \mathcal{N}\left(-x_2\frac{\tau_{12}}{\tau_{11}}, \tau_{11}^{-1}\right)$ and hence $x_2 \sim \mathcal{N}\left(-x_1\frac{\tau_{12}}{\tau_{22}}, \tau_{22}^{-1}\right)$.

I iterated these two equations 100 times and took each of these states as samples. This required approximately 944 flops. On figure 4.1 the result is shown. As seen there is still some correlation, without discussing how to estimate the correlation length, between the samples. The empirical mean is

$$\begin{bmatrix} -0.1831 \\ -0.0900 \end{bmatrix} \tag{4.27}$$

and the empirical covariance

$$\begin{bmatrix} 0.8308 & 0.6333 \\ 0.6333 & 0.8303 \end{bmatrix} \tag{4.28}$$

The Gibbs method is in particular good when it is easy to draw from the conditional distribution, like generating samples from a Gaussian or another pseudo random generator.

4.4.2 Metropolis sampling

Metropolis sampling is a Markov chain methods which looks a lot like rejection sampling. The method is used to produce samples $\left\{\boldsymbol{\omega}^{(i)}\right\}_{i=1}^{N_s}$, the problem is that the samples generated is in general correlated. The Metropolis method also use a sampling distribution which in this case is used to propose a new state based on the previous. Furthermore we have a rejection/acceptance mechanism.

The idea is to propose a new state by the distribution $p_p\left(\boldsymbol{\omega}' \left| \boldsymbol{\omega}^{(n)}\right.\right)$, we can maybe evaluate this up to a multiplicative constant i.e. $q_p\left(\boldsymbol{\omega}' \left| \boldsymbol{\omega}^{(n)}\right.\right)$. We then accept the sample generated from this distribution by $a^{(n+1)} = \min\left(1, \frac{q\left(\boldsymbol{\omega}'\right) q_p\left(\boldsymbol{\omega}^n \left| \boldsymbol{\omega}'\right.\right)}{q\left(\boldsymbol{\omega}^{(n)}\right) q_p\left(\boldsymbol{\omega}' \left| \boldsymbol{\omega}^{(n)}\right.\right)}\right)$ where $q\left(\boldsymbol{\omega}'\right)$ is the non normalized distribution of interest. To decide whether the proposed sample is accepted, a random uniformly distributed number is generated on the interval $[0; 1]$. If the number is lower than $a^{(n+1)}$ the sample is accepted. If not accepted the new sample instead becomes the previous.

Invariability and ergodicity depends much on the proposal distribution. If the proposal distribution is symmetric it can easily be shown that the Markov chain fulfills detailed balance. Furthermore if the proposing distribution independent of the iteration number can propose all possible states, the chain is regular and hence ergodic. If this is only true for some iterations the chain is only irreducible but still ergodic.

One problem with Metropolis sampling is it very easy exhibit *random walk*. Random walk arises when the proposal distribution is very narrow compared to the distribution of interest, this give rise to the proposed state not being so far away from the previous. If one tries to make the proposal distribution less narrow it leads to rejection of too many states. So there exists a optimal tradeoff between these two.

There exists some rule of thumb: if the random walk brings a new sample a distance ϵ a way from the previous state, and the maximal characteristic distance in the distribution of interest is L we can expect to take at least $R = (L/\epsilon)^2$ iterations to get an independent sample.

Example: 2D-Gaussian with Gaussian proposal

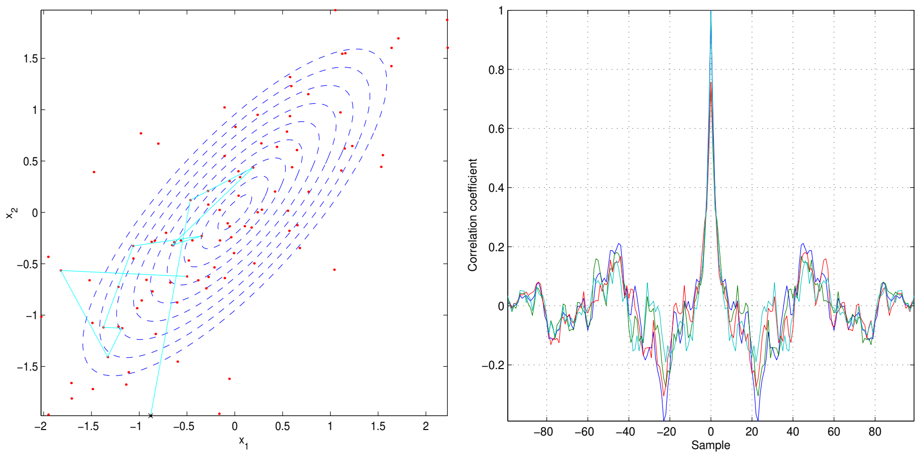
I tried to use the Metropolis algorithm on a 2D-Gaussian with covariance

$$\begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} \tag{4.29}$$

and zero mean. The proposal distribution had isotrop covariance

$$\begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} \tag{4.30}$$

i.e. with variance equal to the smallest eigenvalue of the true covariance. The mean of the proposal is set to the last accepted state. This yields an acceptance of 68%. The largest eigenvalue of the covariance is 1.8 this yields $R = 1.8/0.2 = 9$, by the rule of thumb since the characteristic distance is the largest standard deviation. On figure 4.2.a every 10^{th} state out of 1000 is plotted, which by the rule of thumb are independent samples. The first 10 of these are connected i.e. every 10^{th} state out of the first 100 are connected.



(a) (b)

Figure 4.2: a: Metropolis sampling from a 2D Gaussian with variance 1 in both directions and correlation $\rho = 0.8$. The contours are plotted with equal logarithmic density difference. The proposal distribution was also Gaussian with isotrop covariance variance 0.2 w.i.z. the smallest eigenvalue of the true covariance yielding an acceptance of 68%. The largest eigenvalue is 1.8 hence the approximate distance between independent samples $9 = 1.8/0.2$. The initial sample is marked by a x. Every 10^{th} state is taken as a sample. The 10 first samples are connected by the trajectory i.e. every 10^{th} state of 100. b: correlation between every 10^{th} sample. As seen there is still some correlation between each sample. The three plots are correlation from each direction and there cross correlation.

On figure 4.2.b we see that there still exists some correlation and i.e. some dependence between every 10^{th}. From the 100 samples, every 10^{th} out of 1000 states, the empirical covariance was calculated

$$\begin{bmatrix} 0.9258 & 0.6781 \\ 0.6781 & 0.8670 \end{bmatrix} \tag{4.31}$$

and the empirical mean

$$\begin{bmatrix} -0.1473 \\ -0.1058 \end{bmatrix} \tag{4.32}$$

As seen the estimates are only approximately equal to the true covariance. The evaluation of 100 samples i.e. 1000 states required approximately 36356-flops whereas 100 simple draws from a pseudo random generator only required 885-flops with the rotation and these are nearly independent.

4.4.3 Hybrid Monte Carlo sampling

The Hybrid Monte Carlo method [15], [13] is build upon the "Stochastic Dynamic Method", but it includes a Metropolis acceptance update preventing the stochastic dynamic method to bias the samples. The bias is a consequence of inexact simulation. The major advantage of Hybrid Monte Carlo is the use of gradient information, leading the Markov chain towards areas with higher density. The hope is to suppress the random walk behaviour. The method is inspired from simulation of physical systems. Distributions in statistics can often be taken into this framework, in particular continuous distributions can be simulated provided that there derivatives exists. The tradition in physics is to represent the canonical distribution by its energy function

$$p(\boldsymbol{\omega}) = \frac{1}{z_E} \exp - E(\boldsymbol{\omega}) \tag{4.33}$$

where $E(\boldsymbol{\omega})$ is the energy function. All distributions that are non zero can be taken into this form.

The state variables $\boldsymbol{\omega}$ can be seen as positions of some imaginary particles. The gradient of the energy with respect to this position can then be seen as the force acting on the particles. So we can make a simulation of how these imaginary particles position evolve over time.

To have the particles move they have to be assigned some kinetic energy. For that reason we introduce a parallel energy system

$$p(\mathbf{p}) = \frac{1}{z_K} \exp - K(\mathbf{p}) \tag{4.34}$$

where $\mathbf{p}$ represent the momenta of each particle. The total energy can then be written

$$H(\boldsymbol{\omega}, \mathbf{p}) = E(\boldsymbol{\omega}) + K(\mathbf{p}) \tag{4.35}$$

The distribution of *phase space* i.e. of $(\boldsymbol{\omega}, \mathbf{p})$ can be written

$$\begin{aligned} p(\boldsymbol{\omega}, \mathbf{p}) &= \frac{1}{z_H} \exp - H(\boldsymbol{\omega}, \mathbf{p}) && (4.36) \\ &= \left[\frac{1}{z_E} \exp - E(\boldsymbol{\omega})\right] \left[\frac{1}{z_K} \exp - K(\mathbf{p})\right] && (4.37) \\ &= p(\boldsymbol{\omega})\, p(\mathbf{p}) && (4.38) \end{aligned}$$

so the momenta are independent of the position. So when the simulation of the dynamics is finished we should calculate the marginal distribution of $\boldsymbol{\omega}$ because we are interested in $p(\boldsymbol{\omega})$. As seen the system of interest is decoupled from the invented system, so we can just throw away the $\mathbf{p}$ part of each state calculated.

The distribution of the invented momenta are usually calculated as exponential of the kinetic energy

$$p(\mathbf{p}) = \frac{1}{z_E} \exp -\frac{1}{2} \sum_i \frac{p_i^2}{m_i} \tag{4.39}$$

$$= \frac{1}{\prod_i \sqrt{2\pi m_i}} \exp -\frac{1}{2} \sum_i \frac{p_i^2}{m_i} \tag{4.40}$$

i.e. Gaussian where the variances is the masses m_i of the imaginary particles. In many cases we do not have any prior knowledge of the scales of the different directions so often the masses are chosen to unity.

When simulating the dynamic we are trying to solve the differential equations

$$\frac{d\omega_i}{dt} = +\frac{\partial H}{\partial p_i} = p_i \tag{4.41}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial \omega_i} = -\frac{\partial E}{\partial \omega_i} \tag{4.42}$$

The total energy is unchanged when time progress since

$$\frac{dH}{dt} = \sum_i \frac{\partial H}{\partial \omega_i}\frac{d\omega_i}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i}{dt} \tag{4.43}$$

$$= \sum_i \frac{\partial H}{\partial \omega_i}\frac{\partial H}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial \omega_i} = 0 \tag{4.44}$$

Due to the energy conservation, generating samples from the canonical distribution by following the dynamics will not be ergodic. For this reason a stochastic update is introduced. The stochastic update should be capable of changing the total energy to any particular energy so the method becomes ergodic. This can be carried out by following the dynamics for some time and then drawing some new momenta from the conditional distribution of the momenta given the positions. This can be seen as a Gibbs update and hence this leaves the distribution invariant. Since it is possible during the dynamics update to "move" all the potential energy into kinetic energy and then make a stochastic update that results in zero kinetic energy we can achieve zero total energy. In the other end we can under the distribution of the momenta draw these so the kinetic energy approaches infinity and hence the total energy. Furthermore we could imaging that following the dynamics at a specific total energy would not be ergodic in the sense, that the initial position before the simulation is confined in a region surrounded by "energy barriers" that for this specific energy is impossible to pass, and hence makes it impossible to visit all states with this energy. The stochastic update makes all energy levels possible and all barriers possible to pass, hence the simulation is ergodic.

To make the method usable it should also leave the distribution invariant which is satisfied if detailed balance holds. This can be accomplished by first deciding whether to simulate forward in time or backward in time. Since the dynamics are time reversal this means

$$\omega_i' = \mathcal{F}(\omega_i, p_i) \tag{4.45}$$
$$p_i' = \mathcal{G}(\omega_i, p_i) \tag{4.46}$$

if we run forward in time, then if we run backward in time from the end state, the result should become

$$\omega_i = \mathcal{F}(\omega_i', p_i') \tag{4.47}$$
$$p_i = \mathcal{G}(\omega_i', p_i') \tag{4.48}$$

where $\mathcal{F}$ and $\mathcal{G}$ are the transformation performed by the dynamics.

Detailed balance can then be verified by seeing that if we start in one state with total energy $H(\boldsymbol{\omega}, \mathbf{p}) = H_0$ and simulate the dynamics until we reach a new state which will also have energy $H(\boldsymbol{\omega}', \mathbf{p}') = H_0$ due to energy conservation, then the probability of ending in that state is $\frac{1}{2}\delta V' \frac{1}{z_{H_0}} \exp -H_0$, the half coming from the direction of simulation and the volume is the phase space volume [1] after the transformation. If we instead started in the state $(\boldsymbol{\omega}', \mathbf{p}')$ then the probability of ending in the state $(\boldsymbol{\omega}, \mathbf{p})$ would be $\frac{1}{2}\delta V \frac{1}{z_{H_0}} \exp -H_0$. If the phase space volumes δV and $\delta V'$ are equal after the transformations, for either back or forward solution, then detailed balance holds. This can be examined by the divergence of the transformation from the current state to

[1] A change in phase space volume can be seen as if we have a lot of states spread around in phase space we can calculate the volume they fill, we can then apply our transformation, in this case the dynamics, and afterwards calculate the new volume. If this is different then the phase space volume has changed

the new state. The divergence is

$$\sum_i \frac{\partial}{\partial \omega_i}\left(\frac{d\omega_i}{dt}\right) + \frac{\partial}{\partial p_i}\left(\frac{dp_i}{dt}\right) = \sum_i \frac{\partial^2 H}{\partial \omega_i \partial p_i} - \frac{\partial^2 H}{\partial p_i \partial \omega_i} = 0 \tag{4.49}$$

which show that there is no volume change, and hence detailed balance holds.

To solve the dynamic equations we have to discretize them. The discretization should also fulfill ergodicity and leave the distribution invariant. The ergodicity is assured by the same argument as the continuous solution, but to assure invariance the discretization scheme should still satisfy detailed balance. This can be accomplished by constructing a discretization scheme that preserves phase space volume, is time reversal and does conserve energy. The first property is nearly satisfied for all kind of discretization schemes since the total energy consists of two parts that are mutual independent and hence all the led in the divergence is them self zero. The property of time reversibility can be solved by using *leapfrog* discretization. One *leapfrog* step calculates $\mathbf{p}\,(t+\delta t)$ and $\boldsymbol{\omega}\,(t+\delta t)$ by

$$\begin{aligned} \mathbf{p}(t+\frac{\delta t}{2}) &= \mathbf{p}(t) - \frac{\delta t}{2}\left.\frac{\partial E\,(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}}\right|_{\boldsymbol{\omega}=\boldsymbol{\omega}(t)} && (4.50)\\ \boldsymbol{\omega}(t+\delta t) &= \boldsymbol{\omega}(t) + \delta t\, \mathbf{p}(t+\frac{\delta t}{2}) && (4.51)\\ \mathbf{p}(t+\delta t) &= \mathbf{p}(t+\frac{\delta t}{2}) - \frac{\delta t}{2}\left.\frac{\partial E\,(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}}\right|_{\boldsymbol{\omega}=\boldsymbol{\omega}(t+\delta t)} && (4.52) \end{aligned}$$

It is easily seen that if we solve these equations by using δt with negative sign they will be exact reversible of using a positive sign. The first property of preserving phase space volume is seen to hold for each of the three steps individually. This is easily seen because all transformations that occurs to a variable is independent of the variable it self. Because of dicretization errors we can not be sure that the total energy is preserved and detailed balance does not hold

$$\begin{aligned} &\frac{1}{2}\delta V \exp -H\,(\boldsymbol{\omega},\mathbf{p}) \exp -\left[H\,(\boldsymbol{\omega}',\mathbf{p}') - H\,(\boldsymbol{\omega},\mathbf{p})\right] \neq \\ &\frac{1}{2}\delta V \exp -H\,(\boldsymbol{\omega}',\mathbf{p}') \exp -\left[H\,(\boldsymbol{\omega},\mathbf{p}) - H\,(\boldsymbol{\omega}',\mathbf{p}')\right] && (4.53)\\ &\exp -H\,(\boldsymbol{\omega}',\mathbf{p}') \neq \exp -H\,(\boldsymbol{\omega},\mathbf{p}) && (4.54) \end{aligned}$$

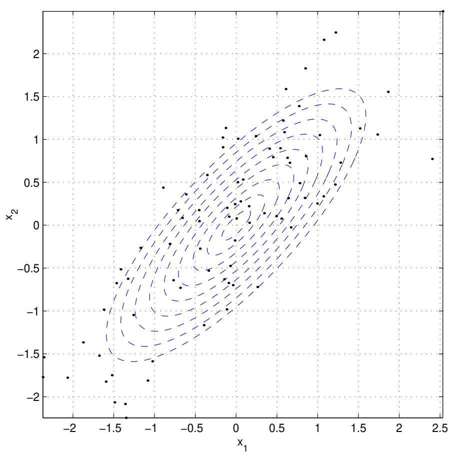
By taking the result of the simulation as a proposal and rejecting and accepting as in the metropolis algorithm, detailed balance is satisfied

$$
\begin{aligned}
&\frac{1}{2}\delta V \exp -H\left(\boldsymbol{\omega},\mathbf{p}\right)\min\left(1,\exp -\left[H\left(\boldsymbol{\omega}',\mathbf{p}'\right)-H\left(\boldsymbol{\omega},\mathbf{p}\right)\right]\right)= \\
&\qquad \frac{1}{2}\delta V \exp -H\left(\boldsymbol{\omega}',\mathbf{p}'\right)\min\left(1,\exp -\left[H\left(\boldsymbol{\omega},\mathbf{p}\right)-H\left(\boldsymbol{\omega}',\mathbf{p}'\right)\right]\right) \\
&\Downarrow \\
&\min\left(\exp -H\left(\boldsymbol{\omega},\mathbf{p}\right),\exp -H\left(\boldsymbol{\omega}',\mathbf{p}'\right)\right)= \\
&\qquad \min\left(\exp -H\left(\boldsymbol{\omega}',\mathbf{p}'\right),\exp -H\left(\boldsymbol{\omega},\mathbf{p}\right)\right)
\end{aligned}
$$

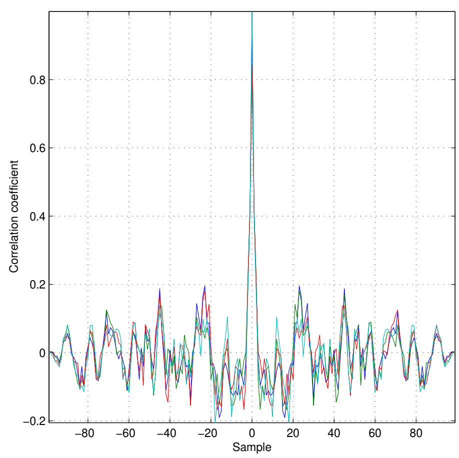
All the different steps in the Hybrid Monte Carlo method can now be written down as an algorithm. For a nice pseudo algorithm see [13].

4.4.4 Example: 2D-Gaussian by Hybrid Monte Carlo

I tried to use the Hybrid Monte Carlo algorithm on the 2D-Gaussian used as example for the Gibbs and Metropolis algorithm. By the use of gradient information random walk should be suppressed. The step length used to simulate the dynamics should be selected between the smallest standard deviation σ_{min} and two times the smallest standard deviation. The smallest standard deviation is typically smaller than the aligned axis standard deviation. This choice is motivated by the fact that we have suppressed random walk and hence we can expect to move away from the previous state by a length of the step size. The larger steps the larger rejection rate so as in the metropolis algorithm there exists a tradeoff. The largest standard deviation compared to the smallest tells us how long we should simulate.



(a)



(b)

Figure 4.3: a: result of drawing 100 samples from a 2D-Gaussian with variances 1 and a correlation coefficient of $\rho = .8$ by Hybrid Monte Carlo. The step size used was $\delta t = 0.6 \simeq 1.5\sigma_{\min}$ and the number of leapfrog steps in each of the 100 draws was $\frac{\sigma_{max}}{\sigma_{min}} = 3$ this yielded an acceptance of 82 i.e. 82 nearly independent samples. b: Correlation coefficients of the samples drawn for each direction, and their cross correlation coefficient. The generation required 26526-flops.

In the case of a correlation coefficient of $\rho = 0.8$ the ratio is $\frac{\sigma_{max}}{\sigma_{min}} = \sqrt{\frac{1.8}{0.2}} = 3$. What we see is the problem scales only proportional to the ratio, whereas in the Metropolis algorithm it scales with the quadratic ratio. On figure

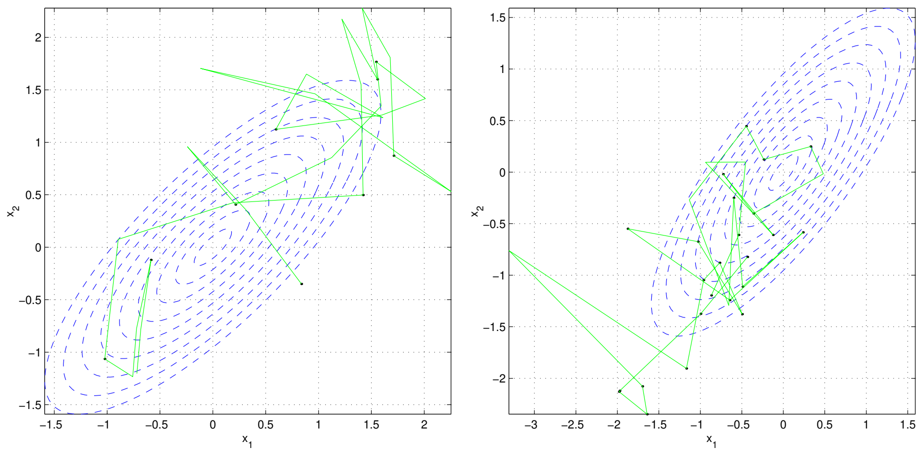


Figure 4.4: a: 10 draws from the 2D-Gaussian with correlation coefficient $\rho = 0.8$ and 3 leapfrog steps between each draw. This resulted in 9 nearly independent draws. The dots shows the samples. This result should be compared to b: here 30 draws was drawn with one leapfrog step between each i.e. the Langevin Monte Carlo. The computer effort is the same as in a. 23-samples was accepted but the independence scales as for the Metropolis algorithm since the space explored by a random walk, so we have 9 samples between the independent samples. So out of the 30 samples we only have 3-4 independent samples. This is seen by the under representation of the upper half of the distribution.

4.3 100 samples was drawn with an acceptance of 82% this required 300 leapfrog steps. The correlation between each sample is fairly low. This can be explained by figure 4.4.a where we see how far away each three leapfrog steps come in the distribution. One of the three leapfrog steps in figure 4.4.a was rejected so the traces to the next sample consists of 6 leapfrog steps, which of course is not true, the trace should be taken back to the previous accepted state. Figure 4.4.a should be compared to figure 4.4.b where each proposal consists of a single leapfrog step, what is called *Langevin Monte Carlo*. As seen the distribution is explored by a random walk, for that reason the problem scales by the quadratic ratio of the largest to the smallest standard deviation. In this case it would require approximately 900 leapfrog steps to obtain 100 samples.

The 100 samples required by the Hybrid Monte Carlo with 3 leapfrog steps between each resulted in an empirical covariance of

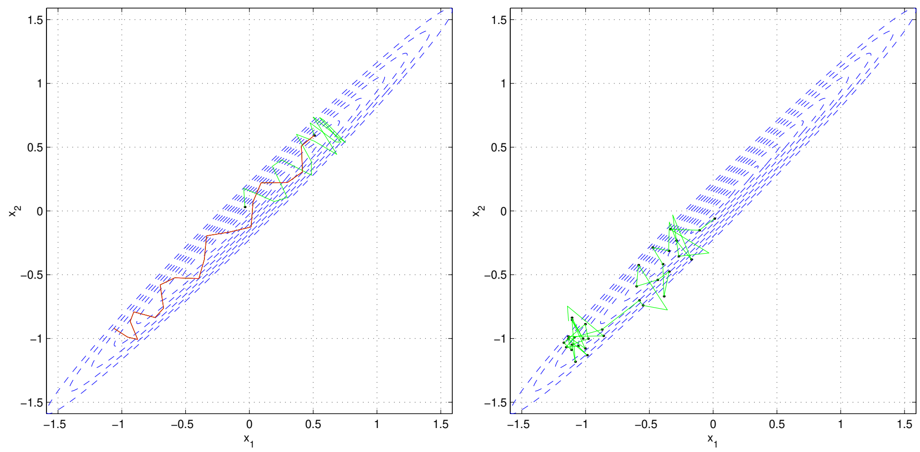


Figure 4.5: a: 2 draws from a 2D-Gaussian with correlation coefficient $\rho = 0.99$ and 20 leapfrog steps between each draw. The stepsize used was 0.15 i.e. 1.5 times larger than the smallest standard deviation. This resulted in 2 independent draws since the distance that should be traveled only was 15 steps i.e. the ratio between the largest and smallest standard deviation . The dots shows the samples. The zig-zag curves shows the 20-leapfrog steps between each sample. This result should be compared to b: here 40 draws was drawn with one leapfrog step between each i.e. the Langevin Monte Carlo. The computer effort is the same as in a. 31-samples was accepted but the independence scales as for the Metropolis i.e. with the squared ratio between the largest and smallest standard deviation, which in this case is 199!!! this is due to exploration by a random walk, so we have only 1 independent sample. This is seen by the under representation of the upper half of the distribution.

$$\begin{bmatrix} 1.1899 & 1.0191 \\ 1.0191 & 1.2300 \end{bmatrix} \tag{4.55}$$

and the empirical mean

$$\begin{bmatrix} 0.0249 \\ 0.0566 \end{bmatrix} \tag{4.56}$$

The difference is even larger in the case seen on figure 4.5, where the distribution know have a correlation coefficient of $\rho = 0.99$. Here two independent samples figure 4.5.a was obtained by using in total 40 leapfrog steps. The ratio between the largest and smallest standard deviation is approximately 15 so the samples should be independent. On figure 4.5.b the same number of leapfrog steps was used by the Langevin Monte Carlo method, as seen the distribution is clearly examined by a random walk. The squared ratio is 199 so obtaining 2 samples by this method would require approximately 400 leapfrog steps or ten times as many as we used by Hybrid Monte Carlo.

5. Calculating the Model Evidence: Theory

As we saw in the previous chapter, Bayesian decisions are based on the evidence[1] $p\left(H\left|\mathbf{D}\right.\right)$. The model H is a description of the model i.e. parameter relations, canonical form of the noise expression, parameters that are unique for the decision e.t.c. All the parameters that are not a part of the hypothesis is integrated "out". What we mean by parameters that are not a part of the hypothesis is, we normally translates the hypothesis into whether some parameters belongs to a certain subspace or not, the remaining part of the parameters are integrated "out". The decision problems that will be addressed here are decisions that are based on which model rather than which parameters. In these kind of setups we choose the model that yields the highest evidence. So the problem is to solve

$$p\left(M_i\left|\mathbf{D}\right.\right) = \frac{p\left(M_i\right)p\left(\mathbf{D}\left|M_i\right.\right)}{\sum_{i'} p\left(M_{i'}\right)p\left(\mathbf{D}\left|M_{i'}\right.\right)} \tag{5.1}$$

where

$$p\left(\mathbf{D}\left|M_i\right.\right) = \int p\left(\mathbf{D}\left|\boldsymbol{\omega}', M_i\right.\right)p\left(\boldsymbol{\omega}'\right)d\boldsymbol{\omega}' \tag{5.2}$$

In some simple models 5.2 can be solved analytic. In others it must be solved by numerical integration. The problem of solving 5.2 is closely related to calculating the "Free energy" of a system. Various numerical methods to solve 5.2 builds on sampling methods. These methods builds on the fact that 5.2 is the normalizing constant z in

$$p\left(\boldsymbol{\omega}\left|\mathbf{D}, M_i\right.\right) = \frac{1}{z}p\left(\mathbf{D}\left|\boldsymbol{\omega}, M_i\right.\right)p\left(\boldsymbol{\omega}\right) \tag{5.3}$$

For that reason these methods are said to "simulate Normalizing Constants". In the following three different approaches to calculate the evidence will be outlined: Importance Sampling, Bridge Sampling and Path Sampling.

5.1 Evidence by Importance Sampling

The Importance Sampling method 4.2, used to calculate expectations of a quantity of interest, is straightforward to use when simulating normalizing

[1] This is my definition, D. Mackay has defined it as $p\left(\mathbf{D}\left|H\right.\right)$

constants. As described in 4.2 we use a sampling distribution[2] $\widetilde{p}(\boldsymbol{\omega}\,|\mathbf{D}, M)$ that can be sampled from and is entirely known and hence normalized, which act as an approximation to the true posterior. The unnormalized posterior is $q(\boldsymbol{\omega}\,|\mathbf{D}, M) = p(\mathbf{D}\,|\boldsymbol{\omega}, M)\,p(\boldsymbol{\omega})$, the normalizing constant is the aim. In 4.2 we introduced the weights $W(\boldsymbol{\omega}) = \frac{q(\boldsymbol{\omega}|\mathbf{D},M)}{\widetilde{p}(\boldsymbol{\omega}|\mathbf{D},M)}$, if we calculate the mean of this quantity with respect to $\widetilde{p}(\boldsymbol{\omega}\,|\mathbf{D}, M)$ this would yield the normalizing constant z

$$
\begin{aligned}
z &= \int \widetilde{p}(\boldsymbol{\omega}\,|\mathbf{D}, M)\,W(\boldsymbol{\omega})\,d\boldsymbol{\omega} && (5.4)\\
&= \int \widetilde{p}(\boldsymbol{\omega}\,|\mathbf{D}, M)\,\frac{q(\boldsymbol{\omega}\,|\mathbf{D}, M)}{\widetilde{p}(\boldsymbol{\omega}\,|\mathbf{D}, M)}d\boldsymbol{\omega} && (5.5)\\
&= \int q(\boldsymbol{\omega}\,|\mathbf{D}, M)\,d\boldsymbol{\omega} && (5.6)
\end{aligned}
$$

A Monte Carlo estimate is then

$$\widehat{z} = \frac{1}{N_s}\sum_{i=1}^{N_s} W(\boldsymbol{\omega}_i) \qquad (5.7)$$

where $\boldsymbol{\omega}_i$ is drawn from $\widetilde{p}(\boldsymbol{\omega}\,|\mathbf{D}, M)$.

This method is straightforward but the variance of the estimate is often high, specifically in high dimensions. This is because the approximating distribution does not match the true distribution.

5.1.1 Approximating the normalizing constant to a 1D Gaussian

This short example shows how hard it is to approximate the normalizing constant to

$$q(x) = \exp -\frac{1}{2}x^2 \qquad (5.8)$$

by Importance Sampling. On figure 5.1 the result from using both a Gaussian with variance 2 and 10000 as sampling distributions is shown. The true result is

$$z = \int q(x)\,dx = \sqrt{2\pi} \qquad (5.9)$$

As seen the estimate becomes more accurate as we get more samples. The accuracy is very low compared to the amount of samples, and if we go to higher dimensions this would increase dramatically.

[2] In these sampling setups I denote the approximations to distribution by adding a tilde, and I name the Normalized densities by p and unnormalized by q

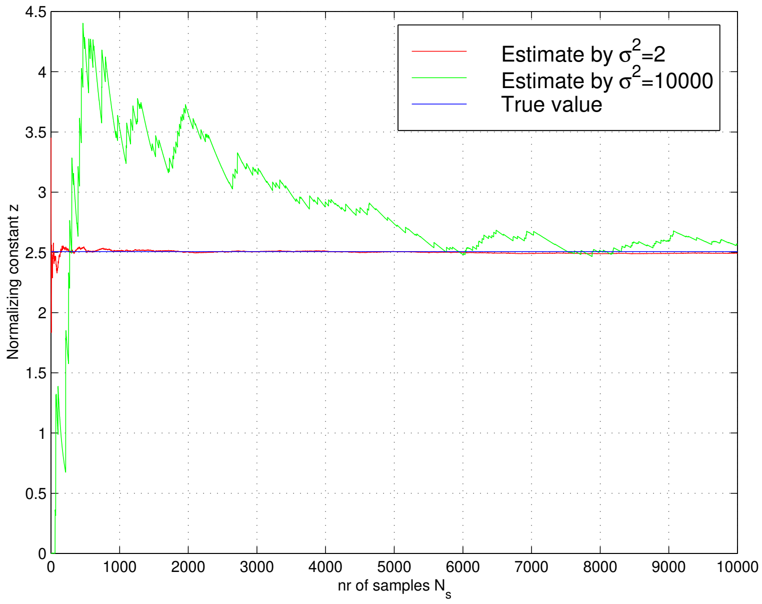


Figure 5.1: Importance sampling estimates of the normalizing constant $z = \sqrt{2\pi}$ to a Gaussian distribution. The sampling distributions are also Gaussian with mean zero but with variance 2 and 10000 respectively

5.2 Evidence by Bridge Sampling methods

In the context of Bridge Sampling methods for calculating the evidence we have "acceptance ratio methods" [2] see also [15] , "Harmonic rule" [17] and "Reciprocal Importance Sampling" [6] and [20]. The common framework for all these methods are described in [14] and partly in [1]. The name Bridge sampling was suggested in [14].

The common framework for Bridge Sampling builds on estimating evidence of models ratios as

$$\frac{p\left(\mathbf{D}\left|M_m\right.\right)}{p\left(\mathbf{D}\left|M_n\right.\right)} = \frac{z_m}{z_n} \tag{5.10}$$

Bridge Sampling builds on sampling technics where we are able to draw samples from $q\left(\boldsymbol{\omega}\left|\mathbf{D},M_i\right.\right) = z_i p\left(\boldsymbol{\omega}\left|\mathbf{D},M_i\right.\right)$ i.e. a distribution that is not normalized. It is very easy to show that

$$\frac{z_m}{z_n} = \frac{E_{\boldsymbol{\omega}|\mathbf{D},M_n}\left[q\left(\boldsymbol{\omega}\left|\mathbf{D},M_m\right.\right)\alpha\left(\boldsymbol{\omega}\right)\right]}{E_{\boldsymbol{\omega}|\mathbf{D},M_m}\left[q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)\right]} \tag{5.11}$$

This is bye the fact that

$$\frac{z_m}{z_n} = \frac{z_m}{z_n}\frac{E_{\boldsymbol{\omega}|\mathbf{D},M_m}\left[q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)\right]}{E_{\boldsymbol{\omega}|\mathbf{D},M_m}\left[q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)\right]} \tag{5.12}$$

$$= \frac{z_m}{z_n}\frac{\int q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)p\left(\boldsymbol{\omega}\left|\mathbf{D},M_m\right.\right)d\boldsymbol{\omega}}{\int q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)p\left(\boldsymbol{\omega}\left|\mathbf{D},M_m\right.\right)d\boldsymbol{\omega}} \tag{5.13}$$

$$= \frac{\int p\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)q\left(\boldsymbol{\omega}\left|\mathbf{D},M_m\right.\right)d\boldsymbol{\omega}}{\int q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)p\left(\boldsymbol{\omega}\left|\mathbf{D},M_m\right.\right)d\boldsymbol{\omega}} \tag{5.14}$$

$$= \frac{E_{\boldsymbol{\omega}|\mathbf{D},M_n}\left[q\left(\boldsymbol{\omega}\left|\mathbf{D},M_m\right.\right)\alpha\left(\boldsymbol{\omega}\right)\right]}{E_{\boldsymbol{\omega}|\mathbf{D},M_m}\left[q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}\right)\right]} \tag{5.15}$$

A Monte Carlo estimate of 5.11 is then

$$\frac{\widehat{z}_m}{\widehat{z}_n} = \frac{\frac{1}{N_n}\sum_{i=1}^{N_n} q\left(\boldsymbol{\omega}_{n,i}\left|\mathbf{D},M_m\right.\right)\alpha\left(\boldsymbol{\omega}_{n,i}\right)}{\frac{1}{N_m}\sum_{i=1}^{N_m} q\left(\boldsymbol{\omega}_{m,i}\left|\mathbf{D},M_n\right.\right)\alpha\left(\boldsymbol{\omega}_{m,i}\right)} \tag{5.16}$$

The function $\alpha(\boldsymbol{\omega})$ act as a function to control the Monte Carlo variance. All the methods mentioned above are all specialized versions of 5.11 with a special choice of $\alpha(\boldsymbol{\omega})$ to minimize the Monte Carlo variance. As with importance sampling it is very important that the two distributions that are used to calculate the ratio are alike, i.e. $q\left(\boldsymbol{\omega}\left|\mathbf{D},M_m\right.\right)$ and $q\left(\boldsymbol{\omega}\left|\mathbf{D},M_n\right.\right)$ are alike. This is not very likely when having two models or systems M_0 and M_1 where we want the ratio $\frac{z_1}{z_0}$ calculated. This problem can be solved by introducing intermediate systems M_θ where $\theta \in [0;1]$ i.e. systems that are characterized by a continuous parameter including the two systems of interest. It is then possible to chain [15] a finite number of these intermediate systems by multiplying all the normalizing constants in the following way

$$\frac{z_1}{z_0} = \frac{z_1}{z_{1-\Delta\theta}}\frac{z_{1-\Delta\theta}}{z_{1-2\Delta\theta}}\frac{z_{1-2\Delta\theta}}{z_{1-3\Delta\theta}}\cdots\frac{z_{2\Delta\theta}}{z_{\Delta\theta}}\frac{z_{\Delta\theta}}{z_0} \tag{5.17}$$

$$= \prod_{i=1}^{1/\Delta\theta}\frac{z_{i\Delta\theta}}{z_{(i-1)\Delta\theta}} \tag{5.18}$$

The hope is then that these intermediate system are more alike. This chaining approach is what is called "Bridge Sampling" [14] because the intermediate systems so to say builds a bridge between the end systems.

5.3 Path Sampling

"Path Sampling" [1] is what the authors (Gelman et al. 1997) call "a natural methodological evolution" of the above described methods. The method is used to calculate the ratio $\log \frac{z_1}{z_0}$ i.e. the logarithm to the ratio defined in the previous section. The method is a generalization of "Bridge Sampling" and hence "Importance Sampling", further more it includes "Thermodynamic Integration" [15] and "Ogata Integration" which are the same.

The method builds on the following

$$\log \frac{z_1}{z_0} = \int_0^1 \frac{d}{d\theta} \log z(\theta)\, d\theta \tag{5.19}$$

$$= \int_0^1 \frac{1}{z(\theta)} \frac{d}{d\theta} \int q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta)\, d\boldsymbol{\omega} d\theta \tag{5.20}$$

where $z(\theta) = \int q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta)\, d\boldsymbol{\omega}$ is used. Assuming the legacy in interchanging the differentiation and integration

$$\begin{aligned}
\log \frac{z_1}{z_0} &= \int_0^1 \frac{1}{z(\theta)} \int \frac{d}{d\theta} q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta)\, d\boldsymbol{\omega} d\theta \\
&= \int_0^1 \frac{1}{z(\theta)} \int q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta) \frac{d}{d\theta} \log q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta)\, d\boldsymbol{\omega} d\theta \\
&= \int_0^1 E_{\boldsymbol{\omega}|\mathbf{D},M_\theta} \left[\frac{d}{d\theta} \log q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta) \right] d\theta
\end{aligned} \tag{5.21}$$

A Monte Carlo estimate is then

$$\log \frac{\widehat{z_1}}{\widehat{z_0}} = \int_0^1 \left(\frac{1}{N_{\theta,s}} \sum_{i=1}^{N_{\theta,s}} U(\boldsymbol{\omega}_i,\theta) \right) d\theta \tag{5.22}$$

where $U(\boldsymbol{\omega},\theta) = \frac{d}{d\theta} \log q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta)$ and $\boldsymbol{\omega}_i \sim p(\boldsymbol{\omega}\,|\mathbf{D},M_\theta)$.

The integration over θ can also be performed by a Monte Carlo integration where the different θ are drawn from $p(\theta)$, which should be thought of as a kind of distribution that tells where most samples should be spent i.e. where there exists largest variance in the sampling path. We could then consider the common space $(\theta, \boldsymbol{\omega})$ as from where we draw samples. We could then draw a sample from $p(\theta)$ and then from $p(\boldsymbol{\omega}\,|\mathbf{D},M_\theta)$ to get the pair $(\theta, \boldsymbol{\omega})$. The Monte Carlo estimate would then become

$$\log \frac{\widehat{z_1}}{\widehat{z_0}} = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{U(\boldsymbol{\omega}_i,\theta_i)}{p(\theta_i)} \tag{5.23}$$

The advantage of this method over "Bridge Sampling" is we have one summand on a log scale that have to converge, instead of two summands on the normal probability scale, which in general should be more stable.

5.3.1 Connection to Thermodynamic Integration

The connection to "Thermodynamic Integration" is obvious, since the free energy or Gibbs energy between two systems in state α_1 and α_0 with the same temperature T is

$$F(T, \alpha_1) - F(T, \alpha_0) = \int_{\alpha_0}^{\alpha_1} E_{\boldsymbol{\omega}|T,\alpha} \left[\frac{\partial H(\boldsymbol{\omega},\alpha)}{\partial \alpha} \right] d\alpha \tag{5.24}$$

where $H(\boldsymbol{\omega},\alpha)$ is the Hamiltonian or energy function of the system. "Thermodynamic Integration" also works between system with different energy or a combination of state and energy.

5.3.2 Connection to Bridge Sampling

The connection to "Bridge Sampling" shows that "Path Sampling" really is "a natural methodology evolution" of the former. The proof here is a little different from the one in [1].

In the section about "Bridge Sampling" we derived 5.17. If we take the logarithm to this we have

$$\log \frac{z_1}{z_0} = \sum_{i=1}^{1/\Delta\theta} \log \frac{z_{i\Delta\theta}}{z_{(i-1)\Delta\theta}} \tag{5.25}$$

if we make the limit where $\Delta\theta \to 0$ then

$$\lim_{\Delta\theta \to 0} \left[\frac{\log z_{i\Delta\theta} - \log z_{(i-1)\Delta\theta}}{\Delta\theta} \right] = \frac{d \log z_\theta}{d\theta} \tag{5.26}$$

and then we have

$$\lim_{\Delta\theta \to 0} \sum_{i=1}^{1/\Delta\theta} \log \frac{z_{i\Delta\theta}}{z_{(i-1)\Delta\theta}} = \int_0^1 \frac{d \log z_\theta}{d\theta} d\theta \tag{5.27}$$

which is the same as what we started out with in equation 5.19 q.e.d.

5.4 Choosing the intermediate systems

The intermediate systems acts as a kind of path between the two systems of interest. The need for these intermediate systems is to decrease the Monte Carlo variance of 5.23. A good path is characterized by a huge match between two distributions with succeeding θ. Or said in other words the gradient

$U(\boldsymbol{\omega},\theta)$ or the change of the intermediate systems should be as small as possible. In [1] a lot of effort has been used to carry out the optimal path if the sampling density $p(\theta)$ is uniform. But since the result depends on the unknown normalizing constants z_1 and z_0 the result is very hard to use in theory. So to do something we use the temparated intermediate systems [15], also denoted the geometric path

$$q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta) = q(\boldsymbol{\omega}\,|\mathbf{D},M_0)^{1-\theta}\, q(\boldsymbol{\omega}\,|\mathbf{D},M_1)^{\theta} \tag{5.28}$$

which in general is suboptimal [1]. The choice of this path is easy to implement if one uses the prior of the parameters as the reference system M_0 i.e. $q(\boldsymbol{\omega}\,|\mathbf{D},M_0) \equiv p(\boldsymbol{\omega}\,|M)$, and since $q(\boldsymbol{\omega}\,|\mathbf{D},M_1) \equiv p(\mathbf{D}\,|\boldsymbol{\omega},M)\, p(\boldsymbol{\omega}\,|M)$ the geometric path 5.28 reduces to

$$q(\boldsymbol{\omega}\,|\mathbf{D},M_\theta) = p(\boldsymbol{\omega}\,|M)\, p(\mathbf{D}\,|\boldsymbol{\omega},M)^{\theta} \tag{5.29}$$

This looks like an obvious choice since we have designed the prior $p(\boldsymbol{\omega}\,|M)$ to have some of the properties that the system of interest also have. So by gradually adding more and more of the likelihood $p(\mathbf{D}\,|\boldsymbol{\omega},M)$ to the prior by θ, we can then hope that the change is small. Another benefit is we often know z_0 which equals 1 if the prior is normalized, so we can get the absolute evidence, instead of a relative one.

5.5 Choice of sampling density

The choice of sampling density $p(\theta)$ also acts as a way to control the accuracy of the solution. Of course the choice of the sampling distribution interact with the results from above. If we use the optimal path based on a uniform sampling density, we can only add in Monte Carlo variance by using another sampling density. But the other way around, if we use a suboptimal path, what the geometric path in general is, we can use a sampling density different from the uniform one to absorb some of the variance added in by the suboptimal path. The optimal sampling density [1] for a given (possible suboptimal) path is

$$p(\theta) = \frac{\sqrt{E_{\boldsymbol{\omega}|\mathbf{D},M_\theta}\left[U(\boldsymbol{\omega},\theta)^2\right]}}{\int_0^1 \sqrt{E_{\boldsymbol{\omega}|\mathbf{D},M_{\theta'}}\left[U(\boldsymbol{\omega},\theta')^2\right]}d\theta'} \tag{5.30}$$

We see 5.30 corresponds to $p(\theta)$ squared being the second moment at a fixed value of θ. This choice is obvious, since we then in the Monte Carlo estimate

5.23 weights each point by the corresponding standard deviation at that θ location. But to get the optimal sampling distribution we have to evaluate two possible impossible integrals. Instead it is possible to build an iterative scheme. But even if we could get the optimal sampling density for a given suboptimal path, we would not in general achieve the same minimal Monte Carlo variance as if we had an optimal path. This is because the structure of the optimal path can be of such complexity, that it is impossible to choose an arbitrary path and then correct it by a sampling density. But the other way around, it is possible to choose an arbitrary sampling density and then correct the optimal path based on the uniform distribution.

But how can we a priori choose a good sampling density when knowing the path ? A heuristic way to chose the sampling density is to think of $U(\boldsymbol{\omega},\theta)$ which with the geometric path is the energy of the likelihood. The distribution of this energy is broader the lower theta i.e. more variant since theta plays the role of inverse temperature. This can also be verified by thinking of θ as a parameter that adds in examples to the posterior, the more examples the more narrow a posterior. So since the sampling density play the role of second moment of $U(\boldsymbol{\omega},\theta)$ we can from the above conclude that $p(\theta) \sim I_{\theta\in[0;1]}\theta^{-1}$. But this is an improper distribution. So instead I suggest using $p(\theta) \sim I_{\theta\in[0;1]}\theta^{-k}$ where $k \in\,]-\infty;1[$ so

$$p(\theta) = \frac{\theta^{-k}}{1-k} I_{\theta\in[0;1]} \tag{5.31}$$

So we will use this sampling density when using the geometric path.

6. Regression models

The topic of regression is of major interest when one wants to model densities like $p(\mathbf{y}\,|\mathbf{x})$, i.e. only modelling the output of a process given the input. The major advantages in Bayesian regression compared to Maximum likelihood is the fact that we do not get any over fit of the model parameters if we select a reasonable prior. In the following two model types will be described: the linear and artificial neural net regression models. Both models will be used in a prediction setup. The methods used is Maximum-Likelihood, Penalized Maximum-Likelihood and the Bayes solutions approximated by a parameterized simpler model which can lead to closed form expressions, sampling and an ensemble approximation. Furthermore we want to select between the different linear models and in the artificial neural net setup between the nets with different complexity. This is accomplished by the Maximum Likelihood or a penalized version or in in the Bayesian context by the evidence / model likelihood.

6.1 The regression model

Both in frequentist and Bayes regression, a model of how the inputs relates to the outputs must be proposed. If the model $\mathbf{f}\,(\mathbf{x};\mathbf{w})$ has parameters given in a vector $\mathbf{w}$, the outputs $\mathbf{y}^{(i)}$ from a single input vector $\mathbf{x}^{(i)}$ can be described by

$$\mathbf{y}^{(i)} = \mathbf{f}\left(\mathbf{x}^{(i)};\mathbf{w}\right) + \boldsymbol{\epsilon}^{(i)} \tag{6.1}$$

where $\boldsymbol{\epsilon}^{(i)}$ are the noise contributions to each output. These noise contributions are modeled by some multivariate distribution $f\left(\boldsymbol{\epsilon}^{(i)}\,|\boldsymbol{\varpi}\right)$, chosen a priori, given some other parameters $\boldsymbol{\varpi}$. Given this noise model and the relational model $\mathbf{f}$, it is easy to derive the likelihood for the model parameters $\boldsymbol{\omega}^T = \left[\mathbf{w}^T, \boldsymbol{\varpi}^T\right]$ given a single input output pair $\left\{\mathbf{x}^{(i)};\mathbf{y}^{(i)}\right\}$

$$f\left(\mathbf{y}^{(i)}\,\middle|\mathbf{x}^{(i)},\boldsymbol{\omega},\mathbf{f}\right) = f\left(\mathbf{f}\left(\mathbf{x}^{(i)};\mathbf{w}\right) - \mathbf{y}^{(i)}\,|\boldsymbol{\varpi}\right) \tag{6.2}$$

From this, the likelihood over a whole data set $\mathcal{D} = \{\mathbf{X};\mathbf{Y}\}$ where $\mathbf{X} = \left[\mathbf{x}^{(1)},\mathbf{x}^{(2)},\ldots,\mathbf{x}^{(N)}\right]$ and $\mathbf{Y} = \left[\mathbf{y}^{(1)},\mathbf{y}^{(2)},\ldots,\mathbf{y}^{(N)}\right]$, can be derived

$$f\left(\mathbf{Y}\,|\mathbf{X},\boldsymbol{\omega},\mathbf{f}\right) = \prod_{i=1}^{N} f\left(\mathbf{f}\left(\mathbf{x}^{(i)};\mathbf{w}\right) - \mathbf{y}^{(i)}\,|\boldsymbol{\varpi}\right) \tag{6.3}$$

The traditional Maximum-Likelihood approach can be applied by maximizing 6.3 with respect to all the model parameters $\boldsymbol{\omega}$. The Bayesian aim is the predictive distribution $p\left(\mathbf{y}^{(N+1)}\left|\mathbf{x}^{(N+1)}, \mathcal{D}, \mathbf{f}\right.\right)$,

$$p\left(\mathbf{y}^{(N+1)}\left|\mathbf{x}^{(N+1)}, \mathcal{D}, \mathbf{f}\right.\right) = \int f\left(\mathbf{y}^{(N+1)}\left|\mathbf{x}^{(N+1)}, \boldsymbol{\omega}', \mathbf{f}\right.\right) k\left(\boldsymbol{\omega}'\left|\mathcal{D}, \mathbf{f}\right.\right) d\boldsymbol{\omega}' \tag{6.4}$$

i.e. the distribution of the possible outputs $\mathbf{y}^{(N+1)}$, when the model is faced with a new input $\mathbf{x}^{(N+1)}$ based on the training data $\mathcal{D}$ and the model $\mathbf{f}$. The posterior $k\left(\boldsymbol{\omega}\left|\mathcal{D}, \mathbf{f}\right.\right)$ can only be accomplished through some kind of prior knowledge $\pi\left(\boldsymbol{\omega}\left|\mathbf{f}\right.\right)$ about the model parameters $\boldsymbol{\omega}$. If this can be accomplished, then the posterior can be derived from Bayes' rule

$$k\left(\boldsymbol{\omega}\left|\mathcal{D}, \mathbf{f}\right.\right) = \frac{f\left(\mathbf{Y}\left|\mathbf{X}, \boldsymbol{\omega}, \mathbf{f}\right.\right) \pi\left(\boldsymbol{\omega}\left|\mathbf{f}\right.\right)}{\int f\left(\mathbf{Y}\left|\mathbf{X}, \boldsymbol{\omega}', \mathbf{f}\right.\right) \pi\left(\boldsymbol{\omega}'\left|\mathbf{f}\right.\right) d\boldsymbol{\omega}'} \tag{6.5}$$

The denominator $d\left(\mathbf{Y}\left|\mathbf{X}, \mathbf{f}\right.\right) = \int f\left(\mathbf{Y}\left|\mathbf{X}, \boldsymbol{\omega}', \mathbf{f}\right.\right) \pi\left(\boldsymbol{\omega}'\left|\mathbf{f}\right.\right) d\boldsymbol{\omega}'$ in 6.5 is what I call the model likelihood, i.e. the marginalized likelihood with respect to all parameters except the model $\mathbf{f}$. The probability of the model given data is then

$$e\left(\mathbf{f}\left|\mathcal{D}\right.\right) = \frac{d\left(\mathbf{Y}\left|\mathbf{X}, \mathbf{f}\right.\right) \pi\left(\mathbf{f}\right)}{\int d\left(\mathbf{Y}\left|\mathbf{X}, \mathbf{f}'\right.\right) \pi\left(\mathbf{f}'\right) d\mathbf{f}'} \tag{6.6}$$

What we in particular want, is to select between different models by comparing their probability given some data. But if we a priori do not have any preferences towards any of the models, i.e. $\pi\left(\mathbf{f}\right)$ is uniform, we could instead compare $d\left(\mathbf{Y}\left|\mathbf{X}, \mathbf{f}\right.\right)$ the likelihood of the models, and select the one attaining the highest model likelihood

6.1.1 The noise model

The noise model should, like a prior, express some a priori knowledge about the noise. If one does not know any thing a priori about the noise, the noise model should express this. The noise model is often chosen to be Gaussian without further edo. This choice is not always obvious a priori, the hypotheses underlying this distribution can be formulated as by Hagen :

- each ε_{ij} consist of an infinite amount of independent infinitesimal errors, all existing on the same scale
- the probability that ε_{ij} is above or beneath some mean value are equal.
- all values, defined on the real-axis, are probable.

The hypothesis follow direct from the central limit theorem.

These hypothesis can not always be met, but even though the noise model is rhetorical chosen to be Gaussian, maybe because of its mathematical convenience. But the noise is actual only Gaussian if we select a specific variance, corresponding to the scale, that we expect all the infinitesimal errors to be on. A more, from my point of view, plausible assumption, is assuming that the infinitesimal errors come on different scales with some kind of distribution. This assumption would lead to some kind of compound noise model. If one assumes the noise to be Gaussian and the inverse noise variance (sometimes called precision) to be distributed according to a gamma distribution, this would yield a compound noise distribution that belongs to the family of t-distributions. This corresponds much to the assumption Bayesians do when using a prior on the noise variance.

6.2 The Linear Model

The linear model is probably one of the most used regression models. The reason for this is obvious when one consider it's analytical tractable solutions. But in a Bayesian context, one quickly reach problems that do not lead to closed form expressions. But for models with a prior that are conjugate to the noise model, some solutions can still lead to closed form expressions.

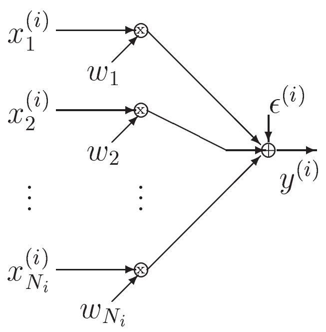


Figure 6.1: The dependencies between inputs and the output, as assumed in the linear model, the output is contaminated with noise $\epsilon^{(i)}$

If we assume the model to be linear, with a single output see figure 6.1, then we will have

$$\mathbf{f}(\mathbf{x};\mathbf{w}) = \mathbf{x}^T\mathbf{w} \tag{6.7}$$

By using a Gaussian noise model, the likelihood becomes

$$f\left(\mathbf{Y}\,|\mathbf{X},\boldsymbol{\omega},\mathbf{f}\right)=\left|\frac{\tau}{2\pi}\right|^{\frac{N}{2}}\exp\left(-\frac{\tau}{2}\left(\mathbf{X}^T\mathbf{w}-\mathbf{Y}\right)^T\left(\mathbf{X}^T\mathbf{w}-\mathbf{Y}\right)\right) \tag{6.8}$$

where τ is the noise precision, and $\boldsymbol{\omega}^T=\left[\mathbf{w}^T,\tau\right]$.

6.2.1 Maximum likelihood for the linear model

The linear model solved by Maximum Likelihood and penalized Maximum Likelihood is a well described topic in statistics [4]. The problem is to maximize 6.8 with respect to $\mathbf{w}$ and τ. This is equivalent to minimize $-\log f\left(\mathbf{Y}\,|\mathbf{X},\boldsymbol{\omega},\mathbf{f}\right)$ with respect to $\mathbf{w}$ and τ. For that reason the derivative with respect to $\mathbf{w}$ is calculated and put equal to zero

$$-\frac{\partial\log f\left(\mathbf{Y}\,|\mathbf{X},\boldsymbol{\omega},\mathbf{f}\right)}{\partial\mathbf{w}}=\tau\left(\mathbf{w}^T\mathbf{X}\mathbf{X}^T-\mathbf{Y}^T\mathbf{X}^T\right)=0 \tag{6.9}$$

which yields the Maximum Likelihood weights

$$\mathbf{w}_{ML}^T=\mathbf{Y}^T\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1} \tag{6.10}$$

The Maximum Likelihood guess of τ can be calculated as

$$-\frac{\partial\log f\left(\mathbf{Y}\,|\mathbf{X},\boldsymbol{\omega},\mathbf{f}\right)}{\partial\tau}= \\ -\frac{1}{2}\tau\left(N-\tau\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}_{ML}\right)^T\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}_{ML}\right)\right)=0 \tag{6.11}$$

which gives

$$\tau_{ML}=\frac{N}{\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}_{ML}\right)^T\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}_{ML}\right)} \tag{6.12}$$

This is only unbiased in the asymptotic where N goes towards infinity. The central guess is accomplished by using the knowledge that an amount of N_i parameters, one weight for each input, already was fitted so

$$\tau_{ML}=\frac{N-N_i}{\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}_{ML}\right)^T\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}_{ML}\right)} \tag{6.13}$$

which is an unbiased guess.

Even though we have the central properties of Maximum Likelihood it will over fit. But to deal with this one has to reduce the effective number of parameters fitted. This can be accomplished in two ways which both regulates the inverse covariance $\mathbf{X}\mathbf{X}^T$. The first is to add an amount α to the diagonal of $\mathbf{XX}^T$, the second is to set some of the directions that spans the covariance $\left(\mathbf{XX}^T\right)^{-1}$ equal to zero. In both cases we then would have an effective number of parameters that are fitted this is in the first case equal to

$$\gamma_{eff} = \sum_{l=1}^{N_i} \frac{\lambda_i}{\lambda_i + \alpha} \tag{6.14}$$

where λ_i is the i^{th} eigenvalue of $\mathbf{XX}^T$. We see that if $\alpha = 0$ then $\gamma_{eff} = N_i$. In the second method the effective number of parameters becomes

$$\gamma_{eff} = \sum_{l=1}^{N_i} \frac{\lambda_i}{\left(\lambda_i^{inv}\right)^{-1}} \tag{6.15}$$

where λ_i^{inv} is the eigenvalues of the covariance $\left(\mathbf{XX}^T\right)^{-1}$ which for some of the values is set to zero. So the effective number of parameters becomes

$$\gamma_{eff} = \sum_{l=1}^{N_i - N_z} \frac{\lambda_i}{\lambda_i} = N_i - N_z \tag{6.16}$$

where N_z is the number of eigenvalues set to zero. The new guess on the noise precision is now

$$\tau_{ML} = \frac{N - \gamma_{eff}}{\left(\mathbf{Y} - \mathbf{X}^T\mathbf{w}_{ML}\right)^T\left(\mathbf{Y} - \mathbf{X}^T\mathbf{w}_{ML}\right)} \tag{6.17}$$

The major problem when penalizing the model is to select how much one should penalize. The way the penalty works is by adding in bias to reduce variance so the overall generalization error falls since this is the sum of these. As we saw this was exactly what Bayes did in the 1D-Gaussian example in section 2.3.2. So Bayes selected the right amount of bias. In the next section when taking linear regression in to a Bayesian context we see α added to the inverse covariance corresponds to putting a Gaussian prior with precision α on the weights.

6.2.2 Analytic Bayesian derivation

In the Analytic derivation, we use a Gaussian prior on the weights $\mathbf{w}$ with a precision $\tau_w = \rho\tau$ and a mean vector equal to the null vector. The noise precision is given a gamma prior with parameters α and β. The prior on the weights corresponds to assuming that small weights are more likely a priori, so trying to drag non important weights towards zero. The noise precision prior parameters can be chosen, such that the prior is vague. This yields

$$\begin{aligned}\pi(\mathbf{w}\,|\rho,\tau) &= \left|\frac{\rho\tau}{2\pi}\right|^{\frac{N_w}{2}} \exp\left(-\frac{\rho\tau}{2}\mathbf{w}^T\mathbf{w}\right)\\ \pi(\tau\,|\alpha,\beta) &= \frac{1}{\Gamma(\alpha)\beta^{\alpha}}\tau^{\alpha-1}\exp-\frac{\tau}{\beta}\end{aligned}$$

where N_w is the number of weights. The total prior then becomes

$$\pi(\boldsymbol{\omega}\,|\rho,\alpha,\beta) = \left|\frac{\rho\tau}{2\pi}\right|^{\frac{N_w}{2}} \frac{1}{\Gamma(\alpha)\beta^{\alpha}}\tau^{\alpha-1}\exp-\tau\left(\frac{\rho}{2}\mathbf{w}^T\mathbf{w}+\frac{1}{\beta}\right) \tag{6.18}$$

which is in the family of normal-gamma distributions. The posterior is then composed by the prior and the likelihood

$$k(\boldsymbol{\omega}\,|\mathcal{D},\rho,\alpha,\beta) = \tag{6.19}$$
$$\frac{(a/2)^{\alpha+\frac{N_c}{2}-\frac{N_w}{2}}\left|\boldsymbol{\Sigma}^{-1}\right|^{1/2}}{\boldsymbol{\Gamma}\left(\alpha+\frac{N}{2}-\frac{N_w}{2}\right)(2\pi)^{\frac{N_w}{2}}}\tau^{\alpha+\frac{N}{2}-1}\exp-\frac{\tau}{2}\left((\mathbf{w}-\mathbf{w}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{w}-\mathbf{w}_0)+a\right)$$

where

$$\begin{aligned}\boldsymbol{\Sigma}^{-1} &= \mathbf{X}\mathbf{X}^T+\rho\mathbb{I}\\ \mathbf{w}_0 &= \boldsymbol{\Sigma}\mathbf{X}^T\mathbf{Y}\\ a &= \frac{2}{\beta}-\mathbf{w}_0^T\boldsymbol{\Sigma}^{-1}\mathbf{w}_0+\mathbf{Y}^T\mathbf{Y}\end{aligned}$$

Now the predictive distribution can be written

$$\begin{aligned}&p\left(y^{(N+1)}\left|\mathbf{x}^{(N+1)},\mathcal{D},\rho,\alpha,\beta,\mathbf{f}\right.\right) =\\ &\qquad\int f\left(y^{(N+1)}\left|\mathbf{x}^{(N+1)},\boldsymbol{\omega}',\mathbf{f}\right.\right)k(\boldsymbol{\omega}'\,|\mathcal{D},\rho,\alpha,\beta)\,d\boldsymbol{\omega}'\end{aligned} \tag{6.20}$$

which leads to, see Appendix A.1

$$\begin{aligned}y^{(n+1)} &\sim t\left(2\alpha+N,\overline{y},\frac{\frac{2}{\beta}\kappa-\overline{y}^2-\kappa\mathbf{Y}^T\mathbf{X}^T\overline{\boldsymbol{\Sigma}}\mathbf{X}\mathbf{Y}+\kappa\mathbf{Y}^T\mathbf{Y}}{2\alpha+N}\right) \qquad (6.21)\\ \overline{\boldsymbol{\Sigma}}^{-1} &= \mathbf{X}\mathbf{X}^T+\mathbf{x}^{(N+1)}\mathbf{x}^{(N+1)^T}+\rho\mathbb{I} = \boldsymbol{\Sigma}^{-1}+\mathbf{x}^{(N+1)}\mathbf{x}^{(N+1)^T}\\ \overline{y} &= \kappa\mathbf{x}^{(n+1)^T}\overline{\boldsymbol{\Sigma}}\mathbf{X}\mathbf{Y}\\ \kappa^{-1} &= 1-\mathbf{x}^{(n+1)^T}\overline{\boldsymbol{\Sigma}}\mathbf{x}^{(n+1)}\end{aligned}$$

i.e. students t-distributed with $2\alpha+N$ degrees of freedom, mean $\overline{y}$ and the covariance is the last parameter.

Selecting the model

The denominator of

$$k\left(\boldsymbol{\omega}\left|\mathcal{D},\rho,\alpha,\beta\right.\right)=\frac{\pi\left(\boldsymbol{\omega}\left|\rho,\alpha,\beta\right.\right)f\left(\mathbf{Y}\left|\mathbf{X},\boldsymbol{\omega},\mathbf{f}\right.\right)}{\int\pi\left(\boldsymbol{\omega}'\left|\rho,\alpha,\beta\right.\right)f\left(\mathbf{Y}\left|\mathbf{X},\boldsymbol{\omega}',\mathbf{f}\right.\right)d\boldsymbol{\omega}'} \tag{6.22}$$

is the evidence of the model. The evidence is the likelihood times the prior marginalized with respect to the parameters in the prior

$$f\left(\mathbf{Y}\left|\mathbf{X},\rho,\alpha,\beta,\mathbf{f}\right.\right)=\int\pi\left(\boldsymbol{\omega}'\left|\rho,\alpha,\beta\right.\right)f\left(\mathbf{Y}\left|\mathbf{X},\boldsymbol{\omega}',\mathbf{f}\right.\right)d\boldsymbol{\omega}' \tag{6.23}$$

This can easily be derived to

$$f\left(\mathbf{Y}\left|\mathbf{X},\rho,\alpha,\beta,\mathbf{f}\right.\right)=\frac{\Gamma\left(\frac{2\alpha+N}{2}\right)\left|\mathbb{I}-\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}\right|}{\Gamma\left(\alpha\right)\left(2\pi\beta^{-1}\right)^{\frac{N}{2}}}\left(1+\frac{\mathbf{Y}^T\left(\mathbb{I}-\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}\right)\mathbf{Y}}{2\beta^{-1}}\right)^{-\frac{2\alpha+N}{2}} \tag{6.24}$$

i.e. $\mathbf{Y}$ is Student's t-distributed

$$\mathbf{Y}\sim t\left(2\beta^{-1},\mathbf{0},\mathbb{I}-\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}\right) \tag{6.25}$$

6.2.3 Monte Carlo Estimate

The Monte Carlo estimate puts the same priors on the various parameters except that τ_w no longer is proportional to τ but it has now it's own prior distribution which also is selected to be of Gamma form with parameters α_w and β_w hence

$$\pi\left(\tau_w\left|\alpha_w,\beta_w\right.\right)=\frac{1}{\Gamma\left(\alpha_w\right)\beta_w^{\alpha_w}}\tau_w^{\alpha_w-1}\exp-\frac{\tau_w}{\beta_w} \tag{6.26}$$

This is a more vague approach than the analytic solution but it does not lead to closed form expressions. But instead it can be solved by Gibbs sampling. To do so we have to find the conditional distributions of the parameters. As a note I do not write in the hyper parameters i.e. the parameters that describes the Gamma distributions, since these are assumed extern given.

First we have the weights conditional distribution

$$p\left(\mathbf{w}\left|\mathbf{Y},\mathbf{X},\tau,\tau_w,\mathbf{f}\right.\right)=\frac{p\left(\mathbf{w}\left|\tau_w\right.\right)f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)}{\int p\left(\mathbf{w}'\left|\tau_w\right.\right)f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w}',\tau,\mathbf{f}\right.\right)d\mathbf{w}'} \tag{6.27}$$

The next conditional distribution is for the noise precision

$$p\left(\tau\left|\mathbf{Y},\mathbf{X},\mathbf{w},\tau_w,\mathbf{f}\right.\right)=\frac{\pi\left(\tau\right)f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)}{\int\pi\left(\tau'\right)f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau',\mathbf{f}\right.\right)d\tau'} \tag{6.28}$$

and at last the the precision on the weights

$$p\left(\tau_w\left|\mathbf{Y},\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)=\frac{\pi\left(\tau_w\right)p\left(\mathbf{w}\left|\tau_w\right.\right)f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)}{\int\pi\left(\tau_w'\right)p\left(\mathbf{w}\left|\tau_w'\right.\right)f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)d\tau_w'} \tag{6.29}$$

which reduces to

$$p\left(\tau_w\left|\mathbf{w}\right.\right)=\frac{\pi\left(\tau_w\right)p\left(\mathbf{w}\left|\tau_w\right.\right)}{\int\pi\left(\tau_w'\right)p\left(\mathbf{w}\left|\tau_w'\right.\right)d\tau_w'} \tag{6.30}$$

since the likelihood is independent of τ_w. The normalization of the conditional distributions can be seen in appendix B by setting $\theta = 1$.

This leads to the following Gibbs updates

- $\mathbf{w}^{(i+1)}\left|\mathbf{Y},\mathbf{X},\tau^{(i)},\tau_w^{(i)},\mathbf{f}\right. \sim \mathcal{N}\left(\mathbf{w}_0,\mathbf{\Sigma}_w^{-1}\right)$
- $\tau^{(i+1)}\left|\mathbf{Y},\mathbf{X},\mathbf{w}^{(i+1)},\mathbf{f}\right. \sim \mathcal{G}\left(a,b\right)$
- $\tau_w^{(i+1)}\left|\mathbf{w}^{(i+1)}\right. \sim \mathcal{G}\left(a_w,b_w\right)$

where

$$\begin{aligned}
&\mathbf{\Sigma}_w^{-1}=\left(\tau^{(i)}\mathbf{X}\mathbf{X}^T+\tau_w^{(i)}\mathbb{I}\right)\\
&\mathbf{w}_0^T=\tau^{(i)}\mathbf{Y}^T\mathbf{X}^T\mathbf{\Sigma}_w\\
&a=\alpha+N/2\\
&b^{-1}=\beta^{-1}+\frac{1}{2}\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}^{(\mathbf{i+1})}\right)^T\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}^{(\mathbf{i+1})}\right)\\
&a_w=\alpha_w-N_i/2\\
&b_w^{-1}=\beta_w^{-1}+\frac{1}{2}\mathbf{w}^{(\mathbf{i+1})^T}\mathbf{w}^{(\mathbf{i+1})}
\end{aligned}$$

By iterating the Gibbs updates one produces samples from $p\left(\mathbf{w},\tau_w,\tau\left|\mathbf{Y},\mathbf{X},\mathbf{f}\right.\right)$. When an amount N_s of independent samples are collected these are used to produce a prediction for a new input x. If the cost is quadratic, the empirical mean of the N_s samples is used for prediction

$$\widehat{y}=\frac{1}{N_s}\sum_{j=1}^{Ns}\mathbf{x}^T\mathbf{w}^{(j)} \tag{6.31}$$

Selecting the model

When one wants to select amongst different models it is necessary to calculate the evidence of the models. The evidence for the linear model, with independence between the noise variance and variance on the weights, can not be calculated analytic, so a sampling scenario have to be carried out. The method used is path sampling with a temperature path. This yields these three conditional distributions for the three parameters of interest

$$p(\mathbf{w}\,|\mathbf{Y},\mathbf{X},\tau,\tau_w,\theta,\mathbf{f}) = \frac{p(\mathbf{w}\,|\tau_w)\,f(\mathbf{Y}\,|\mathbf{X},\mathbf{w},\tau,\mathbf{f})^{\theta}}{\int p(\mathbf{w}'\,|\tau_w)\,f(\mathbf{Y}\,|\mathbf{X},\mathbf{w}',\tau,\mathbf{f})^{\theta}\,d\mathbf{w}'} \quad (6.32)$$

$$p(\tau\,|\mathbf{Y},\mathbf{X},\mathbf{w},\tau_w,\theta,\mathbf{f}) = \frac{\pi(\tau)\,f(\mathbf{Y}\,|\mathbf{X},\mathbf{w},\tau,\mathbf{f})^{\theta}}{\int \pi(\tau')\,f(\mathbf{Y}\,|\mathbf{X},\mathbf{w},\tau',\mathbf{f})^{\theta}\,d\tau'} \quad (6.33)$$

$$p(\tau_w\,|\mathbf{w}) = \frac{\pi(\tau_w)\,p(\mathbf{w}\,|\tau_w)}{\int \pi(\tau'_w)\,p(\mathbf{w}\,|\tau'_w)\,d\tau'_w} \quad (6.34)$$

The solution when normalizing these can be seen in appendix B. Besides drawing the parameters one should draw inverse temperatures from $p(\theta)$. The Gibbs updates now is in two loops

1. $\theta^{(i)} \sim p(\theta)$
2. for $j = 1$ to N_q
   - $\mathbf{w}^{(j+1)}\,\Big|\mathbf{Y},\mathbf{X},\tau^{(j)},\tau_w^{(j)},\theta^{(i)},\mathbf{f} \sim \mathcal{N}\left(\mathbf{w}_0,\mathbf{\Sigma}_w^{-1}\right)$
   - $\tau^{(j+1)}\,\Big|\mathbf{Y},\mathbf{X},\mathbf{w}^{(j+1)},\theta^{(i)},\mathbf{f} \sim \mathcal{G}(a,b)$
   - $\tau_w^{(j+1)}\,\big|\mathbf{w}^{(j+1)} \sim \mathcal{G}(a_w,b_w)$
3. endfor

The inner loop is iterated N_q times, where N_q is selected so the state after these iterations are independent [1] of the state before these iteration. When an appropriate amount N_s of samples $\left(\theta^{(i)},\mathbf{w}^{(i)},\tau^{(i)},\tau_w^{(i)}\right)$ where i indicates the independent states, are collected, the evidence of the model can be estimated according to 5.23.

6.2.4 A toy problem: Selecting the number of inputs

The linear regression model will now be used on a toy problem. The data sets are generated by drawing 20 input vectors each consisting of 10 inputs.

[1] we can of course only guess on independence since this is very hard to measure

So $\mathbf{X}$ is $10x20$. Each of the inputs are independent $\mathcal{N}(0,1)$. The three first inputs are added together to form the outputs $\mathbf{Y}$ for each of the 20 examples. To the output is added Gaussian white noise with standard deviation 0.01. To assure good statistics 100 experiment are performed.

We now ask how many inputs are needed ? We then select the model that are proposed to perform best by the various selecting criterias. Then we measure the performance of the selected models by an independent test set generated so it contains 200 examples. From this we then ask is the selection criteria consistent with the test error.

Maximum Likelihood on toy problem

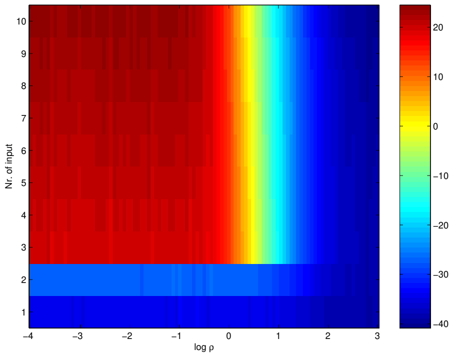
The maximum likelihood estimate of the weights is already derived 6.10.

The results are shown on figure 6.2. In table 6.1 different selection methods are used.

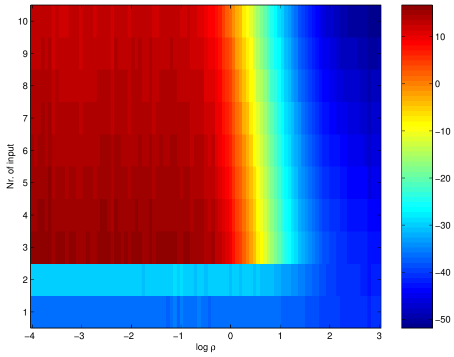
If we select according to the non regularized Maximum Likelihood out left in all the plots, the number of output selected would be $N_i = 10$. But we see on figure 6.2.c that this model over fits the data since the test variance table 6.1 is 0.0219 which should be 0.0100. The penalized methods both select the "true" model. We see on figure 6.2 that the decision area is not homogeneous, this is due to the finite number of times the experiment is replicated.

	$\frac{\rho_{ML}}{10^{-4}}$	inputs	log like.	p. log like.	σ^2_{train}	σ^2_{test}	log pre
ML	0	10	23.7301	12.7301	0.0098	0.0219	0.2764
Reg. ML	1.92	10	24.4891	13.4891	0.0092	0.0191	0.3873
Pen. ML	0	3	19.8886	15.8886	0.0098	0.0118	0.7917
P., r. ML	1.92	3	20.7219	16.7219	0.0090	0.0118	0.7808
Truth		3	17.6720	14.6720	0.0100	0.0100	0.8836

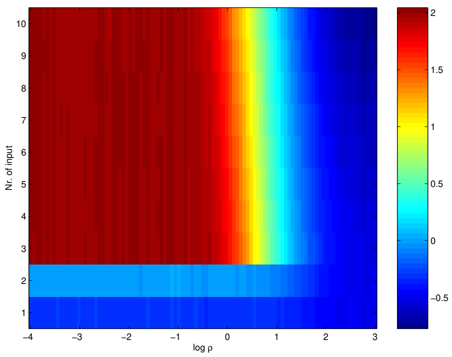
Table 6.1: Four different selection methods based on the Maximum Likelihood principle. All values consists of 100 experiments with 20 training cases and 200 test cases. The ML indicates pure Maximum Likelihood. The Reg. ML indicates that the inverse covariance is reguralized, the maximum log reguralized value is chosen which yields a specific amount of reguralization. Pen. ML means Maximum penalized Likelihood, the penalty is simply the number of weights estimated. P., r. ML means both penalized likelihood and reguralized inverse covariance. Truth is the result when using the true distribution, log likelihood is then the density of 20 "mean" examples i.e. 20 times the log predictive, pen. log. is the same number minus the true number of inputs i.e. 3. The two variances is the true variance and log predictive is $-\frac{1}{2}\log\left(2\pi\sigma^2_{true}\right) - \frac{1}{2}$



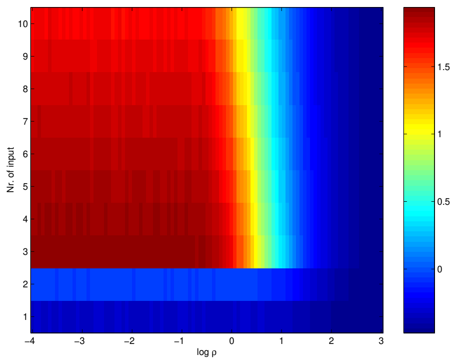
(a) Log likelihood of the different models with different regularization coefficient ρ w.i.z. added to the inverse covariance



(b) Log likelihood minus N_i i.e. penalized likelihood



(c) $-log_{10}$ to the estimated variance on the training set



(d) $-log_{10}$ to the estimated variance on the test set

Figure 6.2: a: likelihood and b: penalized likelihood for selection. c: The training error normalized by the number of examples minus the number of weights. d: The test error normalized by the number of test examples. In all four plots the leftmost value of ρ equals zero i.e. no regularization

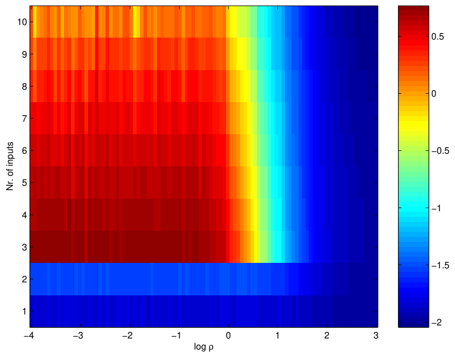


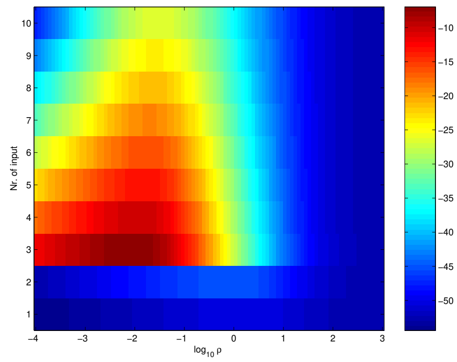
Figure 6.3: The log predictive distribution averaged over 200 test cases and 100 experiments each based on 20 training cases. The true value should be 0.8836. As seen for the true model with 3 inputs we are just below this value with the nearest being 0.7917 see table 6.1, but for increasing number of inputs there exists too many parameters and the model over fits the data.

Analytic Bayes on the toy problem

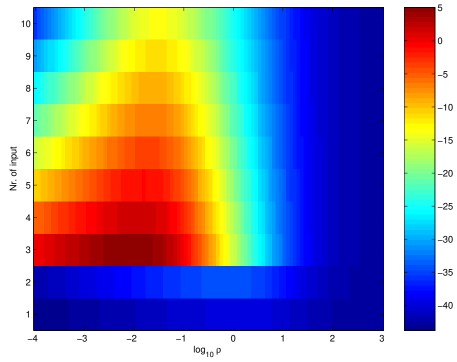
The analytic Bayes solved in 6.2.2 leads to the evidence results seen on figure 6.4. The prior on the noise precision and hence on the precision on the weight prior is chosen very vague. On figure 6.4.a the prior is very vague approaching the non informative Jeffrey prior, on 6.4.b the prior is more constrained which is seen to yield nearly the same result. The constrained prior is actually expressing a kind of a priori "feeling" namely we do not expect that large or low noise levels. The result is quiet surprising, in both setups the model with $N_i = 3$ is selected independent of ρ i.e. the true model is selected independent of ρ. I tried to use some other prior parameters, and the result is nearly independent of these only for very narrow priors the test failed.

The test error figure 6.5 is the negative logarithm of the predictive distribution averaged with respect to all the test examples. This is actually the log-loss part of the $\mathcal{KL}$-distance measure.

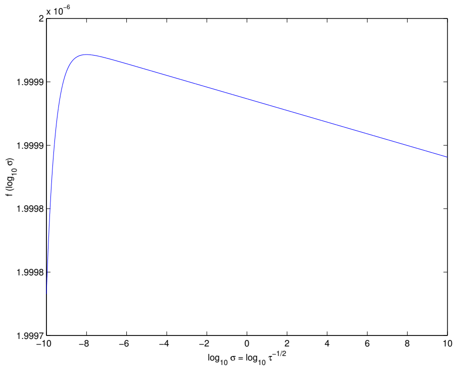
We see that there is a perfect match between the choice of model we would select based on the evidence and the logarithm of the predictions i.e. minus test error. This is because there is no over fitting in the Bayesian framework, hence the training data error should provide the same information as an independent test set. What we also see is when using



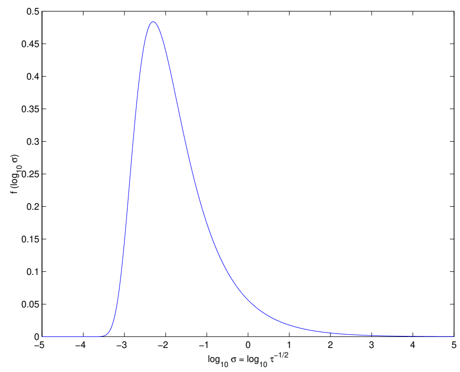
(a) $\alpha = 10^{-6}$,$\beta = 10^{8}/\alpha$



(b) $\alpha = .5$, $\beta = 200/\alpha$



(c) $\alpha = 10^{-6}$, $\beta = 10^{8}/\alpha$



(d) $\alpha = .5$, $\beta = 200/\alpha$

Figure 6.4: On a: the evidence is plotted against ρ by use of the prior on c. b: shows the result produced by using the prior on d.

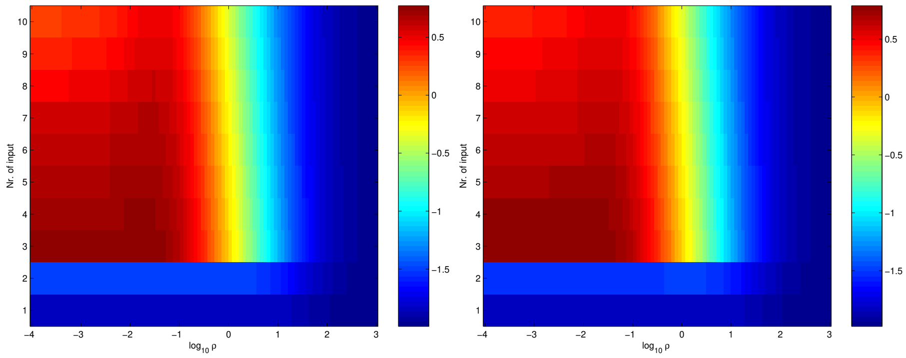


Figure 6.5: a: log predictive distribution when using the vague prior. b: the same but for the more narrow prior. Both figures are made by calculating the predictive distribution based on 20 training data cases averaged over 200 test cases. This experiment is performed 100 times and the log predictive distribution is averaged over these 100 runs. The averaged log-predictive distribution should yield the logarithm of the true distribution averaged with respect to the test-cases i.e. a value of $-\frac{1}{2}\log(2\pi\sigma_{true}^2) - \frac{1}{2} \simeq 0.8836$ since $\sigma_{true} = 0.1$.

Monte Carlo Bayes on the toy problem

In the analytic Bayes setup we imposed the constrained $\tau_w = \rho\tau$, this is of course an unnecessary constrain. What it actually means is, if the noise level raise then the scale of the weights follows proportional to this. But this is a little awkward since the total signal level remains the same and since the signal level consists of the input level times the weights and the noise level we would expect them to be inversely proportional. So no constrains are implied on the problem in this setting, but the price is that we can no longer derive closed form expressions, and we have to specify one more prior.

Since this is a sampling setup we should examine how well we are "getting around" in the distribution of interest. We know if the Markov states are very correlated then the chain should be run for some time before taking a state as a sample. But in general it is very difficult to say anything about how well the samples represent the distribution.

In a setup like this, where we want to calculate the evidence by adjusting the temperature, it is even worse to state anything about the Markov chain because the system is not stationary. In principle we should make an exhaustive sampling epoch at each temperature of interest. But this would be

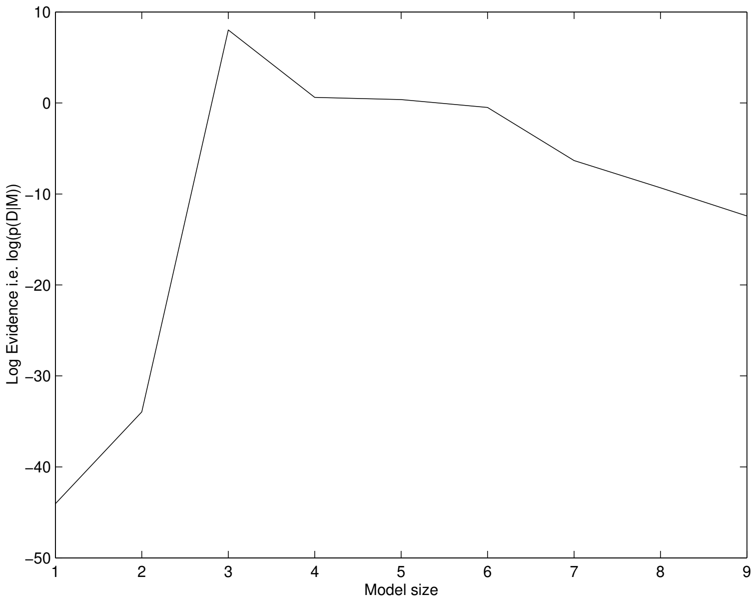
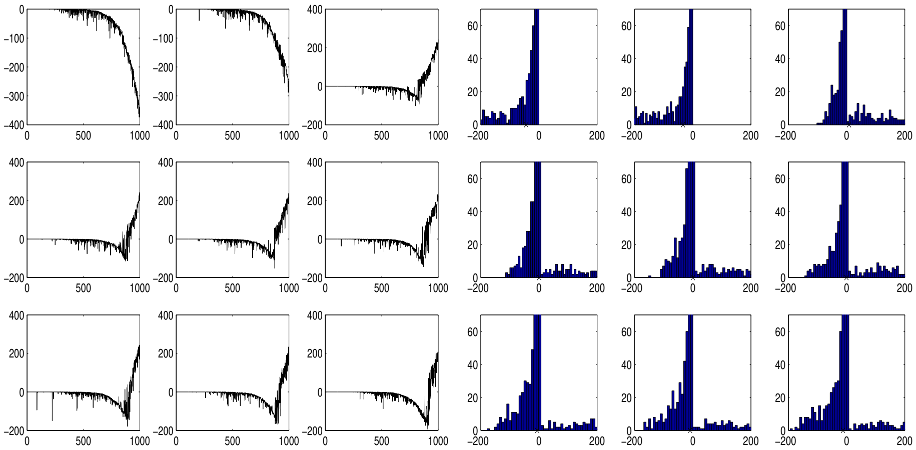


Figure 6.6: On the figure is seen the log evidence i.e. log density of data given the model. The 9 models corresponds to models with 1-9 of the first inputs. The true model is also linear using the 3 first inputs. The figure is constructed from a single data set with 20 examples. The gamma priors on τ and τ_w was given the parameters $\alpha = 1/2$, $\beta = 1/200$ and $\alpha_w = 1/4$, $\beta_w = 1/1600$ respectively.

very time consuming. Instead we make an exhaustive examination at Bayes temperature i.e. $\theta = 1$, we then hope that at higher temperatures it would take shorter or the same time to move around in the distribution. This is motivated by the fact that an increasing temperature flattens out the likelihood, and hence can only make it more simple.

On figure 6.6 is the Monte Carlo estimate of the log evidence for nine models shown. Each model corresponds to a model with 1-9 of the first inputs, the true model using the 3 first inputs. A MCMC simulation for each of the nine models were performed. The parameters of the gamma priors for τ and τ_w was $\alpha = 1/2$, $\beta = 1/200$ and $\alpha_w = 1/4$, $\beta_w = 1/1600$ respectively i.e vague priors. The sampling at different temperatures was performed in decreasing temperature order. So before the simulation started 1000 inverse temperatures was drawn from $p(\theta)$ given by equation 5.31 with $k = 0.9$ which gave the lowest Monte Carlo variance compared to $k = 0.99$, $k = 0.5$ and $k = 0$. After the inverse temperatures was drawn they were sorted in



(a) Inverse temperature θ (b) Sample values

Figure 6.7: a: 1000 values of $U(\boldsymbol{\omega}_i)/p(\theta_i)$ at their respective inverse temperature. See the text for simulation details. b: histograms of the 1000 samples, the mean is the log evidence of that model, the mean is marked by a x

increasing order. The start state at the first temperature which is near inf, was initialized by $\tau = 1$ and $\tau_w = 1$. From this state 3 Gibbs iterations were made taking the third as the sample $\{\theta_0, \boldsymbol{\omega}_0\}$ at that initial temperature. Then the second highest temperature is used and again 3 Gibbs iterations is performed from the previous sample, this goes on with the next temperature until the 1000 samples are obtained.

After this $U(\boldsymbol{\omega}_i)/p(\theta_i)$ is calculated for each of the 1000 samples and for each of the 9 models. On figure 6.7.a the 1000 samples for each of the 9 models is shown as function of inverse temperature. Remembering that the inverse temperature act as the fraction of examples in the likelihood, we see from the third model to the ninth model in the sample interval 800 to 940 corresponding to around 3 to 9 examples, a transition which in physics would have been characterized as a phase transition occurs. The transition is located at still higher θ for larger models i.e. when seeing more and more examples. This is obvious since a linear model with 9 parameters can not be fitted until we have seen as many independent examples. Until this critic temperature where the transition can occurs, their is no evidence for the model which can bee seen on figure 6.8.a where we see that the noise level corresponds to modeling all the signal as noise and at figure 6.8.b we see that this is consistent with the width of the weight distribution, which is

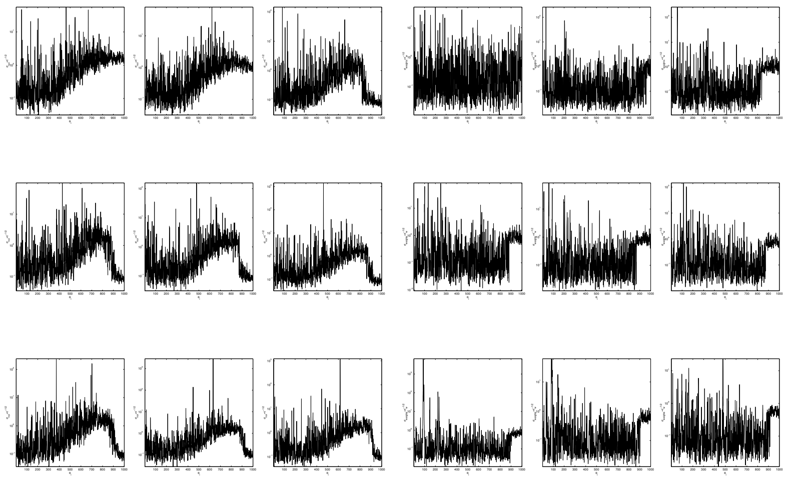


Figure 6.8: On the nine left most figures the noise standard deviation is shown for each of the models. On the nine right most figures the standard deviation of the weights are shown for each of the nine models. All the figures is as function of the inverse temperature.

very small corresponding to small weights i.e. closing down the inputs. So at the critic temperature we have a phase transition between a model which is only noise and a model modeling the true signal and noise. Of course the models larger than three inputs does not totally close down the extra inputs which is the reason that these come out with a slightly lower log evidence. If the model could "see" from the examples that it should close down these extra inputs the evidence of this model would be more the same as the evidence for the true model, but I think it always will be lower since a priori this weight choice is not very likely. For the two models with less than 3 inputs we see no matter the number of examples there will newer be a phase transition, since the true model can never be modeled.

The Monte Carlo variance is of course of major importance. The Monte Carlo estimate is as mentioned the mean of the 1000 $U(\boldsymbol{\omega}_i)/p(\theta_i)$. On figure 6.7.b we can see how much spread there is around each estimated log evidence, the estimate is marked by a $\times$. As described the distribution $p(\theta)$ acts as a second moment stabilizer, by selecting $k = 0.9$ we got a good stabilization. This choice gives rise to a lot of sampling at low θ, why this is "optimal" can be explained by the fact that at this temperature the posterior is very width and hence we have a lot of variation, so we need a lot of samples to get a good estimate of the mean. We can see from figure 6.7.a that it is very important for the estimate when the phase transition occurs, since the value changes drastically at this temperature. This of course depends on

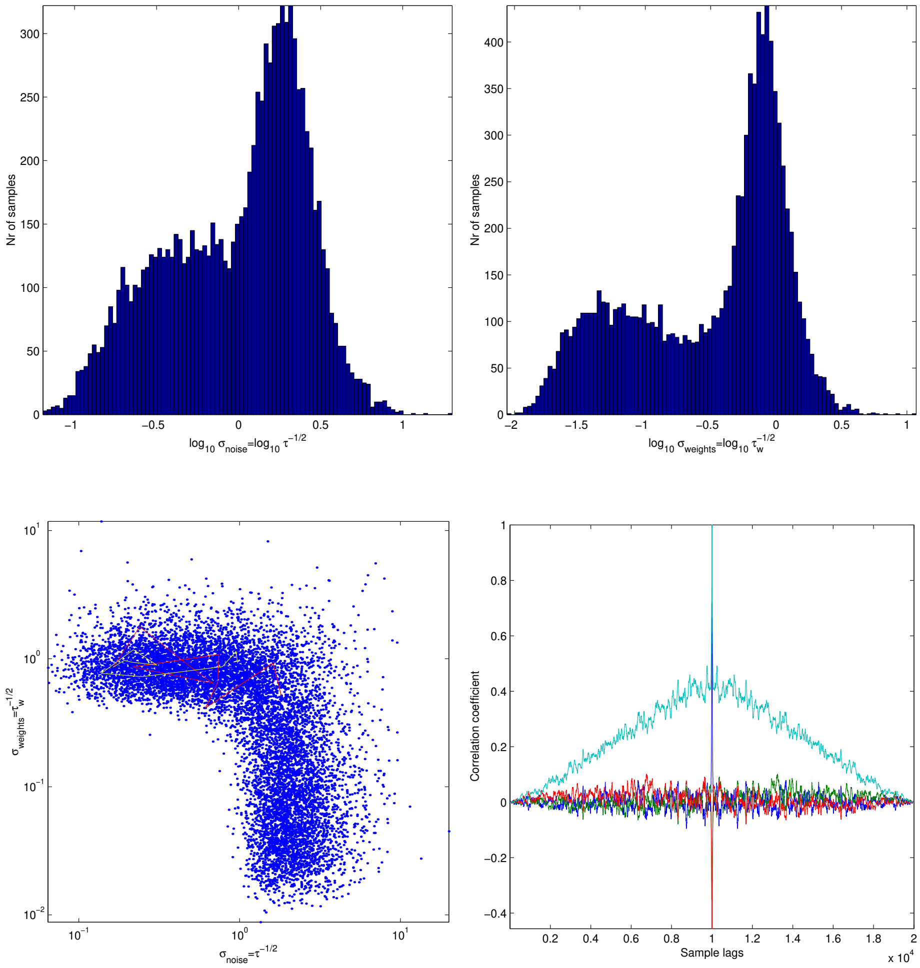


Figure 6.9: a: Marginal distribution of $\sigma_{noise} = \tau$. b: Marginal distribution of $\sigma_{weights} = \tau_w$. c: The joint between $\sigma_{noise} = \tau$ and $\sigma_{weights} = \tau_w$, the traces show how fast the Markov Chain gets around in the distribution, each trace consists of 9 states. d: Correlation and cross correlation between $\sigma_{noise} = \tau$ and $\sigma_{weights} = \tau_w$. The two graphs with negative correlation in zero are the cross correlations which is negative due to the inverse proportional connection between $\sigma_{noise} = \tau$ and $\sigma_{weights} = \tau_w$. The correlation of $\sigma_{noise} = \tau$ is very near a delta function. The correlation of $\sigma_{weights} = \tau_w$ is seen to be very long, we can see that two time constants is playing a role

how well our sampling method comes around in the distribution,- especially at temperatures around the phase transition where both models are likely it is critic. To illustrate how the distribution looks around the phase transition 10000 succeeding states from a Markov Chain, at inverse temperature $\theta = 0.1580$ corresponding to 3.16 examples, was generated. The model used had 4 inputs. The result is shown on figure 6.9. We see 6.9.a,b,c that the joint distribution of $\sigma_{noise} = \tau$ and $\sigma_{weights} = \tau_w$ is bimodal. We see 6.9.c,d that the noise level is sampled quit fast, the trajectories comes around over the whole scale very fast 6.9.c. But because $\sigma_{weights} = \tau_w$ is coupled to $\sigma_{noise} = \tau$ through the weights and if these are moving around by random walk, it will take some time before we can get a new independent $\sigma_{weights} = \tau_w$ sample, this is due to the fact that $\sigma_{weights} = \tau_w$ is the scale of all the weights, hence a form of consensus amongst the weights is required to move $\sigma_{weights} = \tau_w$.

The MCMC-method presented here is a little time consuming and hence the model was not run a 100 times as when using Maximum Likelihood and the analytic solution. I think the result in general cases is worth waiting for, because we do not have to run for many different data set. Unfortunately did I not measure the test and training error of this setup.

6.2.5 Comparison of the three methods

In the previous three sections three different approaches have been used to select how many inputs that are relevant to the linear model. The models with more than 3 inputs can implement the correct model. The Maximum Likelihood solution performed purely. It gave a bias towards the larger models, this is of course due to over fitting. The same over fitting resulted in a too optimistic training error, and too pessimistic test error. The penalized Maximum Likelihood method performed very well. The penalty coming from the AIC-criteria or more precise the PE-criteria. This penalty effectively down weights the over fitting. The test and training error spread is acceptable. This performing so well is not that surprising, the penalty actually introduces bias in the same way as Bayes does in the analytic example of predicting from a 1D-Gaussian with known variance and unknown mean.

The analytic Bayes method performed extreme well even though we implied the constrain that the weight hyperparameter should be proportional to the noise hyperparameter. The test errors was consistent with the model choice. Even for the largest model the test error was very near the true error, which was not the case in any of the maximum likelihood setups.

The MCMC-estimate also selected the true model. The over fitting problem was not discussed but, by the results in the next Chapter, I think no over fitting has occurred since this is not the case for the Feedforward Neural Network regression. But the results with the linear model shows a lot of different problems that makes it hard to estimate the evidence, like the high variance at high temperatures and the phase transition at a critical temperature.

6.3 Artificial Neural Network regression

The Artificial Neural Network has the opportunity to approximate any smooth function arbitrary close by increasing the number of hidden units [3]. This by itself can not assure that we can find that network, but only the existence of such. When we try to infer the function from data the problem becomes probabilistic since the data have a certain accuracy. In such cases the error function determines what we focus on in the distribution of data. By using the quadratic error function we know the output to be modeled is the conditional mean i.e. the mean of the output given the inputs. This is much the same as assuming the noise to be Gaussian and using the KL-distance as error function between the resulting predictive distribution and the true distribution. In this thesis will I use a single layer feed forward neural network with tanh activations, linear outputs and assuming additive Gaussian noise.

6.3.1 The Network Architecture

The formal description is as follows, the outputs are a linear combination of the hidden units and biases

$$\mathbf{f}\left(\mathbf{x}^{(i)};\mathbf{W}_O;\mathbf{W}_I\right) = \mathbf{W}_O^T\mathbf{h}\left(\mathbf{x}^{(i)};\mathbf{W}_I\right) \tag{6.35}$$

where $\mathbf{h}$ is the hidden units i.e.

$$\begin{aligned} h_j\left(\mathbf{x}^{(i)};\mathbf{w}_j\right) &= \tanh \mathbf{w}_j^T\mathbf{x}^{(i)} \;\; j \in [0; N_h - 1] & (6.36)\\ h_{N_h}\left(\mathbf{x}^{(i)};\mathbf{w}_j\right) &= 1 & (6.37)\end{aligned}$$

$\mathbf{w}_j$ being the j^{th} column of $\mathbf{W}_I$ and $\mathbf{x}^{(i)} = \left[x_0^{(i)}, x_1^{(i)}, \ldots, x_{N_h-1}^{(i)}, 1\right]^T$.

From this the likelihood can easily be written down assuming Gaussian noise with the same (inverse squared) level τ on the different outputs

$$p\left(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)}\left|\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)},\mathbf{W}_O,\mathbf{W}_I,\tau\right.\right) = \tag{6.38}$$

$$\left(\frac{\tau}{2\pi}\right)^{\frac{N_O N}{2}} \exp -\frac{\tau}{2}\sum_{i=1}^{N}\left\|\mathbf{y}^{(i)} - \mathbf{f}\left(\mathbf{x}^{(i)};\mathbf{W}_O;\mathbf{W}_I\right)\right\|_2 \tag{6.39}$$

where N_o is the number of outputs and N being the number of examples.

By using the negative logarithm of the likelihood as error function is

$$E\left(\mathbf{W}_O, \mathbf{W}_I, \tau\right) = -\frac{N_o N}{2} \log \frac{\tau}{2\pi} + \frac{\tau}{2} \sum_{i=1}^{N} \left\| \mathbf{y}^{(i)} - \mathbf{f}\left(\mathbf{x}^{(i)}; \mathbf{W}_O; \mathbf{W}_I\right) \right\|_2 \quad (6.40)$$

the derivatives of this can easily be calculated omitting the τ

$$\begin{aligned} \frac{\partial E\left(\mathbf{W}_O, \mathbf{W}_I, \tau\right)}{\partial \mathbf{W}_O^{k'j'}} &= \sum_{k=1}^{N_o} \sum_{i=1}^{N} \delta_k^{(i)} \frac{\partial f_k\left(\mathbf{x}^{(i)}; \mathbf{W}_O; \mathbf{W}_I\right)}{\partial \mathbf{W}_O^{k'j'}} & (6.41) \\ &= \sum_{i=1}^{N} \delta_k^{(i)} h_{j'}\left(\mathbf{x}^{(i)}; \mathbf{w}_j\right) \\ \frac{\partial E\left(\mathbf{W}_O, \mathbf{W}_I, \tau\right)}{\partial \mathbf{W}_I^{j'l'}} &= \sum_{k=1}^{N_o} \sum_{i=1}^{N} \delta_k^{(i)} \frac{\partial f_k\left(\mathbf{x}^{(i)}; \mathbf{W}_O; \mathbf{W}_I\right)}{\partial \mathbf{W}_I^{j'l'}} & (6.42) \\ &= \sum_{k=1}^{N_o} \sum_{i=1}^{N} \delta_k^{(i)} \mathbf{W}_O^{kj'} \frac{\partial h_{j'}\left(\mathbf{x}^{(i)}; \mathbf{W}_O; \mathbf{W}_I\right)}{\partial \mathbf{W}_I^{j'l'}} \\ &= \sum_{k=1}^{N_o} \sum_{i=1}^{N} \delta_k^{(i)} \mathbf{W}_O^{kj'} \left(1 - h_{j'}^2\left(\mathbf{x}^{(i)}; \mathbf{W}_O; \mathbf{W}_I\right)\right) x_{l'}^{(i)} \end{aligned}$$

where $\delta_k^{(i)} = f_k\left(\mathbf{x}^{(i)}; \mathbf{W}_O; \mathbf{W}_I\right) - y_k^{(i)}$ is the backpropagation error from the output.

The gradient is going to be used in all different solutions.

6.3.2 The Maximum Likelihood solution

The Maximum Likelihood solution is to find minima of the error function or more specific the regularized error function called the cost function given by

$$C\left(\mathbf{W}_O, \mathbf{W}_I, \tau\right) = E\left(\mathbf{W}_O, \mathbf{W}_I, \tau\right) + \mathbf{w}^T \mathbf{R} \mathbf{w} \quad (6.43)$$

$\mathbf{w}$ being all the weights arranged in a single vector and $\mathbf{R}$ being the regularizer. In this setup we will only consider diagonal $\mathbf{R}$. The reguralizer trying to prevent from over fitting.

In the simulations performed, existing software [18] was used. The software uses conjugate gradients [3] to find the minima. The software used direct weights from the input to output, this was removed to obtain comparable results.

To select amongst the models will the direct training error, test error on an independent test set and the FPE-criterion to estimate the test error from the training data. Unfortunately we should use the modified selection criteria GPE [3] using the effective number of parameters estimated from the Hessian matrix. The estimates actually only being valid if the network have capacity to implement the true function, but this will in general not be the case due to the fact that infinitely many hidden units should be used to implement an arbitrary continuous function. To compensate from this we should use the GPE-criterion (Generalized Prediction Error) [3]. Unfortunately is the Hessian not available from the software [2] so I use the FPE-criterion

$$E_{test} = \frac{N + N_w}{N - N_w} E_{train} \tag{6.44}$$

6.3.3 The Bayesian solution

The Bayesian solution is to put priors on $\mathbf{w}$ and τ and then aim for the predictive distribution. I will use the same priors as suggested in the literature [16], [3]. These are the conjugate priors as used for the linear model.

$$\pi\left(\mathbf{w}^{(k)}\left|\tau_w^{(k)}\right.\right) = \left|\frac{\tau_w^{(k)}}{2\pi}\right|^{\frac{N_w^{(k)}}{2}} \exp\left(-\frac{\tau_w^{(k)}}{2}\mathbf{w}^{(k)^T}\mathbf{w}^{(\mathbf{k})}\right) \tag{6.45}$$

$$\pi\left(\tau_w^{(k)}\left|\alpha_w^{(k)}, \beta_w^{(k)}\right.\right) = \frac{1}{\Gamma\left(\alpha_w^{(k)}\right)\beta_w^{(k)^{\alpha_w^{(k)}}}}\tau_w^{(k)^{\alpha_w^{(k)}-1}} \exp -\frac{\tau_w^{(k)}}{\beta_w^{(k)}} \tag{6.46}$$

$$\pi\left(\tau\left|\alpha, \beta\right.\right) = \frac{1}{\Gamma(\alpha)\beta^\alpha}\tau^{\alpha-1}\exp -\frac{\tau}{\beta} \tag{6.47}$$

The weights are split into four groups $k \in [1;4]$, $\mathbf{w}^{(k)}$ being the weight in that group and $N_w^{(k)}$ being the number of weights in the group. The four groups are split into: bias to hidden, input to hidden, bias to output and hidden to output. These four groups have their own priors. From this we can formally write the posterior of the parameters, the inverse temperature is included for later use

$$p\left(\tau\left|\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}, \mathbf{w}, \boldsymbol{\theta}\right.\right) = \tag{6.48}$$

[2] I did not have the time to take the weights into Matlab where I have software to calculate the Hessian

$$\frac{p^{\theta}\left(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)}\left|\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)},\mathbf{w},\tau\right.\right)p\left(\tau\right)}{\int p^{\theta}\left(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)}\left|\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)},\mathbf{w},\tau'\right.\right)p\left(\tau'\right)d\tau'}$$

and for the weights

$$p\left(\mathbf{w}\left|\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)},\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)},\tau,\theta\right.\right)= \tag{6.49}$$
$$\frac{p^{\theta}\left(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)}\left|\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)},\mathbf{w},\tau\right.\right)\prod_{k=1}^{4}p\left(\mathbf{w}^{(k)}\left|\tau_{w}^{(k)}\right.\right)}{\int p^{\theta}\left(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)}\left|\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)},\mathbf{w}',\tau\right.\right)\prod_{k=1}^{4}p\left(\mathbf{w}'^{k}\left|\tau_{w}^{(k)}\right.\right)d\mathbf{w}'}$$

and lastly the precision of the weight groups

$$p\left(\tau_{w}^{(k)}\left|\mathbf{w}^{(k)}\right.\right)=\frac{p\left(\mathbf{w}^{(k)}\left|\tau_{w}^{(k)}\right.\right)p\left(\tau_{w}^{(k)}\right)}{\int p\left(\mathbf{w}^{(k)}\left|\tau_{w}'^{(k)}\right.\right)p\left(\tau_{w}'^{(k)}\right)d\tau_{w}'^{(k)}} \tag{6.50}$$

The first and the last posterior can be normalized analytic, since the priors are conjugate priors. The derivation follows closely what we did for the linear model so the results can be found in appendix B. The difficult part is the posterior of the weights and the integral marginalizing the parameters to yield the predictive distribution. Two approaches will be discussed, the first, the ensemble estimate, is approximating the posterior of the weights with a Gaussian distribution given the means of the other parameters. When doing this it is possible to calculate the predictive distribution over the approximation leading to a closed form expression of the predictive distribution. The ensemble method is close related to the evidence approximation by Mackay [10], except that the reestimation formulas has fewer variational parts. The second approach is a sampling scenario. The noise precision and the precision of the weights can be sampled by Gibbs updates given the weights and the weights can be sampled by using the Hybrid Monte Carlo method.

6.3.4 Ensemble estimation

Ensemble learning is a recent developed method in Bayesian learning to approximate the posterior distribution [11], [12], [9]. The method, also called variational approximation, approximates the posterior with a parametric model that assumes independence between some parameters and which can lead to closed form expressions for the predictive distribution. This is carried out by minimizing some measure between the true posterior and the approximation, the measure is often the Kullback-Leibler distance. The difference from this method compared to the evidence framework by MacKay [10], [3] is the approximation around the mean rather than approximating around a peak[3] of the posterior.

In appendix C is the approximation to the posterior for the ANN derived. I will shortly summerize the results. The posterior $p(\mathbf{w}, \tau, \boldsymbol{\alpha} | D, \theta)$ is approximated by

$$q(\mathbf{w}, \tau, \boldsymbol{\alpha} | D, \theta) = q(\mathbf{w} | D, \theta)\, q(\tau | D, \theta)\, q(\boldsymbol{\alpha} | D, \theta) \tag{6.51}$$

The purpose is now to minimize the KL-distance between the true and approximating posterior

$$\begin{aligned} \mathcal{KL}\left(p(\mathbf{w}, \tau, \boldsymbol{\alpha} | D, \theta), q(\mathbf{w}, \tau, \boldsymbol{\alpha} | D, \theta)\right) = \\ -\int q(\mathbf{w}, \tau, \boldsymbol{\alpha} | D, \theta) \log \frac{p(\mathbf{w}, \tau, \boldsymbol{\alpha} | D, \theta)}{q(\mathbf{w}, \tau, \boldsymbol{\alpha} | D, \theta)} d\mathbf{w}, \tau, \boldsymbol{\alpha} \end{aligned} \tag{6.52}$$

The solutions can be solved by iteration, in the m^{th} iteration the results are

$$q^{(m)}(\mathbf{w} | D, \theta) = \sqrt{\left|\frac{\boldsymbol{\Sigma}_w^{(m)^{-1}}}{2\pi}\right|} \exp -\frac{1}{2}\left(\mathbf{w} - \overline{\mathbf{w}}^{(m)}\right)^T \boldsymbol{\Sigma}_w^{(m)^{-1}} \left(\mathbf{w} - \overline{\mathbf{w}}^{(m)}\right) \tag{6.53}$$

where

$$\boldsymbol{\Sigma}_w^{(m)^{-1}} = \left.\frac{\partial^2 E_w(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T}\right|_{\mathbf{w} = \overline{\mathbf{w}}^{(m-1)}} \tag{6.54}$$

and

$$\overline{\mathbf{w}}^{(m)} = \overline{\mathbf{w}}^{(m-1)} - \boldsymbol{\Sigma}_w^{(m)} \left.\frac{\partial E_w(\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w} = \overline{\mathbf{w}}^{(m-1)}} \tag{6.55}$$

where E_w is the exponential of the non linear $q(\mathbf{w} | D, \theta)$ i.e.

$$E_w = -\frac{1}{2}\left(\theta \overline{\tau} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + \sum_{k=1}^{K} \sum_{i=1}^{N_k} \overline{\alpha}_k \left(w_i^{(k)}\right)^2\right) \tag{6.56}$$

the bars over variables indicates the mean of those variables with respect to their approximating posterior. The posterior for τ becomes

$$q^{(m)}(\tau | D, \theta) \propto \tau^{a_\tau - 1} \exp -\tau / b_\tau^{(m)}$$

[3] or several peaks.

i.e. a gamma distribution $\Gamma\left(a_\tau, b_\tau^{(m)}\right)$,where

$$\begin{aligned} a_\tau &= a_{p,\tau} + \frac{N\theta}{2} \\ b_\tau^{(m)} &= \left[b_{p,\tau}^{-1} + \frac{\theta}{2} \sum_{i=1}^{N} \left(\delta_i^2 \left(\overline{\mathbf{w}}^{(m)} \right) + \frac{\tau}{2} \sum_{jj'=1}^{N_w} \left. \frac{\partial^2 \delta_i^2 (\mathbf{w})}{\partial w_j \partial w_{j'}^T} \right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m)}} \Sigma_{jj'}^{(m)} \right) \right]^{-1} \end{aligned}$$

so the mean becomes

$$\overline{\tau}^{(m)} = a_\tau b_\tau^{(m)} \tag{6.57}$$

The posterior approximation for the $\tau_w = \alpha$ in each group indicated by (k), not to confuse with the iteration (m), simply becomes K independent gamma distributions

$$q^{(m)}\left(\alpha_k \left| D, \theta\right.\right) = \frac{1}{\Gamma(a_k)\, b_k^{(m)a_k}} \alpha_k^{a_k - 1} \exp -\alpha_k / b_k^{(m)} \tag{6.58}$$

K being the number of weight groups, so

$$q^{(m)}\left(\alpha \left| D, \theta\right.\right) = \prod_{k=1}^{K} q^{(m)}\left(\alpha_k \left| D, \theta\right.\right)$$

where

$$\begin{aligned} a_k &= a_{p,k} + \frac{N_k}{2} \\ b_k^{(m)^{-1}} &= \left[b_{p,k}^{-1} + \frac{1}{2} \sum_{i=1}^{N_k} \left(\Sigma_{ii}^{(k)(m)} + \overline{w}_i^{(k)(m)^2} \right) \right]^{-1} \end{aligned}$$

where p indicates the prior, so

$$\overline{\alpha}_k^{(m)} = a_k b_k^{(m)} \tag{6.59}$$

Iterating these equations until convergence will yield an approximation to the posterior, an iteration scheme can be seen in appendix C. What we see is the solution is much the same as the normal gradient decent with full hessian and weight decay used in Maximum Likelihood settings. The ensemble method provides the extra re-estimation formulas for the weight decays and noise levels.

Predictive distribution when using the Ensemble estimate

The approximation can be used in two ways directly by integrating the likelihood of a new example with respect to the posterior approximation or the second way by producing samples from the posterior approximation in an importance like fashion.

The first approach will require an approximation of the likelihood of the new example by expanding the network output around the mean weight vector. The integral over the weights is then Gaussian which yields a new Gaussian independent of the weights but dependent on the noise precision. The noise precision can then be marginalized by multiplying the approximation to the posterior of the noise precision on the Gaussian and then integrating with respect to τ.

The sampling approach can be used to produce samples from the posterior, these samples can eventually be used to produce the mean output of the network with respect to the predictive distribution given this new example

$$\begin{aligned}
&\overline{\mathbf{y}}^{(N+1)} \left| \mathbf{x}^{(N+1)} \right. = \\
&\int \mathbf{y}^{(N+1)} p\left(\mathbf{y}^{(N+1)} \left| \mathbf{x}^{(N+1)} \right.\right) d\mathbf{y}^{(N+1)} = \\
&\int \mathbf{y}^{(N+1)} \int p\left(\mathbf{y}^{(N+1)} \left| \mathbf{x}^{(N+1)}, \mathbf{w}, \tau \right.\right) p\left(\mathbf{w}, \tau, \alpha \left| \mathcal{D} \right.\right) d\mathbf{w}, \tau, \alpha d\mathbf{y}^{(N+1)} = \\
&\int\int \mathbf{y}^{(N+1)} p\left(\mathbf{y}^{(N+1)} \left| \mathbf{x}^{(N+1)}, \mathbf{w}, \tau \right.\right) d\mathbf{y}^{(N+1)} p\left(\mathbf{w}, \tau, \alpha \left| \mathcal{D} \right.\right) d\mathbf{w}, \tau, \alpha
\end{aligned} \tag{6.60}$$

by using that the network output is the mean in the Gaussian likelihood of the new example we have

$$\overline{\mathbf{y}}^{(N+1)} \left| \mathbf{x}^{(N+1)} \right. = \int \mathbf{f}\left(\mathbf{x}^{(N+1)}; \mathbf{w}\right) \frac{p\left(\mathbf{w}, \tau, \alpha \left| \mathcal{D} \right.\right)}{q\left(\mathbf{w}, \tau, \alpha \left| \mathcal{D} \right.\right)} q\left(\mathbf{w}, \tau, \alpha \left| \mathcal{D} \right.\right) d\mathbf{w}, \tau, \alpha \tag{6.61}$$

where the multiplication with respect to the approximation is made to obtain the following Monte Carlo estimate

$$\overline{\mathbf{y}}^{(N+1)} \left| \mathbf{x}^{(N+1)} \right. \simeq \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{f}\left(\mathbf{x}^{(N+1)}; \mathbf{w}_i\right) \frac{p\left(\mathbf{w}_i, \tau_i, \alpha_i \left| \mathcal{D} \right.\right)}{q\left(\mathbf{w}_i, \tau_i, \alpha_i \left| \mathcal{D} \right.\right)} \tag{6.62}$$

but since the true posterior $p\left(\mathbf{w}_i, \tau_i, \alpha_i \left| \mathcal{D} \right.\right)$ not can be normalized we should use the following estimate

$$\overline{\mathbf{y}}^{(N+1)} \left| \mathbf{x}^{(N+1)} \right. \simeq \frac{\sum_{i=1}^{N_s} \mathbf{f}\left(\mathbf{x}^{(N+1)}; \mathbf{w}_i\right) \frac{p^*\left(\mathbf{w}_i, \tau_i, \alpha_i \left| \mathcal{D} \right.\right)}{q\left(\mathbf{w}_i, \tau_i, \alpha_i \left| \mathcal{D} \right.\right)}}{\sum_{i=1}^{N_s} \frac{p^*\left(\mathbf{w}_i, \tau_i, \alpha_i \left| \mathcal{D} \right.\right)}{q\left(\mathbf{w}_i, \tau_i, \alpha_i \left| \mathcal{D} \right.\right)}} \tag{6.63}$$

where the * indicates non normalized posterior like

$$p^{*}\left(\mathbf{w},\tau,\alpha\left|\mathcal{D}\right.\right)=p\left(\left\{\mathbf{y}^{(i)}\right\}_{i=1}^{N}\left|\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N},\mathbf{w},\tau\right.\right)p\left(\mathbf{w}\left|\alpha\right.\right)p\left(\alpha\right)p\left(\tau\right) \qquad (6.64)$$

Of course this approximation suffers from the usual problem when using Importance sampling, if the posterior approximation is far from the real posterior the method performs purely.

Ensemble estimate and model selection by Importance sampling

The Importance sampling method used in the previous section to estimate the mean outputs can also be used to estimate the evidence of the model

$$\begin{aligned} p\left(\mathcal{D}\left|M\right.\right) &= \int p\left(\mathbf{w},\tau,\alpha,\mathcal{D}\left|M\right.\right)d\mathbf{w},\tau,\alpha \\ &= \int \frac{p\left(\mathbf{w},\tau,\alpha,\mathcal{D}\left|M\right.\right)}{q\left(\mathbf{w},\tau,\alpha\left|\mathcal{D}\right.\right)}q\left(\mathbf{w},\tau,\alpha\left|\mathcal{D}\right.\right)d\mathbf{w},\tau,\alpha \end{aligned} \qquad (6.65)$$

where the Monte Carlo estimate becomes

$$p\left(\mathcal{D}\left|M\right.\right)\simeq\frac{1}{N_s}\sum_{i=1}^{N_s}\frac{p\left(\mathbf{w}_i,\tau_i,\alpha_i,\mathcal{D}\left|M\right.\right)}{q\left(\mathbf{w}_i,\tau_i,\alpha_i\left|\mathcal{D}\right.\right)} \qquad (6.66)$$

the N_s samples drawn from the posterior approximation. Again this method suffers from the problems detected when using Importance sampling.

6.3.5 Monte Carlo estimation

The Monte Carlo estimate builds on the fact that we are able to draw samples from the true posterior at any (inverse) temperature. Doing this we are able to estimate the predictions and the evidence of the model. The samples can be obtained in a combination of Gibbs-sampling and Hybrid Monte Carlo. The noise precision and the precision on the weights can be obtained by Gibbs sampling given the weights and data, since these are Gamma-distributions. The weights can be obtained by Hybrid Monte Carlo given the two precisions and data.

The posterior of the noise precision and precision on the weights are derived in an equivalent way as in the linear model, see appendix B, the results being

$$p\left(\tau_w^{(k)}\left|\mathbf{w}^{(k)}\right.\right) = \frac{\left({b_w^{(k)}}^{-1}\right)^{a_w^{(k)}}}{\Gamma\left(a_w^{(k)}\right)}{\tau_w^{(k)}}^{a_w^{(k)}-1}\exp -\tau_w^{(k)}{b_w^{(k)}}^{-1} \qquad (6.67)$$

$$
\begin{aligned}
a_w^{(k)} &= \alpha_w^{(k)} - N_w^{(k)}/2 \\
b_w^{(k)^{-1}} &= \beta_w^{(k)^{-1}} + \frac{1}{2}\mathbf{w}^{(k)^T}\mathbf{w}^{(k)}
\end{aligned}
$$

where the index (k) indicates the k^{th} weight group. The posterior for the noise precision

$$
\begin{aligned}
p\left(\tau\left|\mathcal{D},\mathbf{w},\theta\right.\right) &= \frac{b^{\alpha+\theta\frac{NN_0}{2}}}{\Gamma\left(\alpha+\theta N_o\frac{N}{2}\right)}\tau^{\alpha+\theta N_o\frac{N}{2}-1}\exp -\tau b^{-1} && (6.68) \\
b^{-1} &= \beta^{-1} + \frac{\theta}{2}\sum_{i=1}^{N}\left\|\mathbf{y}^{(i)} - \mathbf{f}\left(\mathbf{x}^{(i)};\mathbf{w}\right)\right\|_2 && (6.69)
\end{aligned}
$$

The posterior for the weight can formally be written

$$
\begin{aligned}
p\left(\mathbf{w}\left|\mathcal{D},\tau,\theta\right.\right) &\propto p^{\theta}\left(\left\{\mathbf{y}^{(i)}\right\}_{i=1}^{N}\middle|\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N},\mathbf{w},\tau\right)p\left(\mathbf{w}\middle|\left\{\tau_w^{(k)}\right\}_{k=1}^{K}\right) \\
&\propto \exp -\left(\theta E\left(\mathbf{w},\tau\right) + E_{prior}\left(\mathbf{w}\right)\right) && (6.70)
\end{aligned}
$$

The Hybrid Monte Carlo algorithm makes use of the gradient of the energy function $\frac{\partial E(\mathbf{w},\tau)+E_{prior}(\mathbf{w})}{\partial\mathbf{w}}$ the first part of this was derived in section 6.3 [4]. The last derivative is simply $\frac{\partial E_{prior}(\mathbf{w})}{\partial \mathbf{w}_j^{(k)}} = \mathbf{w}_j^{(k)}\tau_w^{(k)}$ where j indicates the j^{th} weight in the k^{th} group.

By sampling at unit temperature $\theta = 1$ we can obtain samples from the posterior. We can then estimate the mean output given a new input

$$
\begin{aligned}
\overline{\mathbf{y}}^{(N+1)}\left|\mathbf{x}^{(N+1)}\right. &= \int \mathbf{y}^{(N+1)}p\left(\mathbf{y}^{(N+1)}\left|\mathbf{x}^{(N+1)}\right.\right)d\mathbf{y}^{(N+1)} \\
&= \int \mathbf{y}^{(N+1)}\int p\left(\mathbf{y}^{(N+1)}\left|\mathbf{x}^{(N+1)},\mathbf{w},\tau\right.\right)p\left(\mathbf{w},\tau,\alpha\left|\mathcal{D}\right.\right)d\mathbf{w},\tau,\alpha d\mathbf{y}^{(N+1)} \\
&= \int\int \mathbf{y}^{(N+1)}p\left(\mathbf{y}^{(N+1)}\left|\mathbf{x}^{(N+1)},\mathbf{w},\tau\right.\right)d\mathbf{y}^{(N+1)}p\left(\mathbf{w},\tau,\alpha\left|\mathcal{D}\right.\right)d\mathbf{w},\tau,\alpha \\
&= \int \mathbf{f}\left(\mathbf{x}^{(N+1)};\mathbf{w}\right)p\left(\mathbf{w}\left|\mathcal{D}\right.\right)d\mathbf{w} && (6.71)
\end{aligned}
$$

The last integral can be estimated by an Monte carlo estimate

$$
\overline{\mathbf{y}}^{(N+1)}\left|\mathbf{x}^{(N+1)}\right. \simeq \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{f}\left(\mathbf{x}^{(N+1)};\mathbf{w}_i\right) \qquad (6.72)
$$

[4] here was τ omitted so the derivative is the derivative from that section multiplied by $\tau\theta$

where the weights are drawn from the posterior of the weights.

When estimating the log evidence $\log p(\mathcal{D}\,|M)$ we need to sample at a lot of different inverse temperatures θ in the interval from 0 to 1. The θ's are chosen from the distribution $p(\theta)$ in advance and sorted in ascending order. The parameters are then sampled at each (inverse) temperature to make the following Monte Carlo estimate of the evidence 5.23

$$\log p\,\widehat{(\mathcal{D}\,|M)} = \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{U(\boldsymbol{\omega}_i,\theta_i)}{p(\theta_i)} \tag{6.73}$$

where

$$\begin{aligned} U(\mathbf{w},\theta) &= \frac{\partial}{\partial\theta}\log q(\mathbf{w},\tau,\alpha\,|M,\mathcal{D},\theta) \\ &= \frac{\partial}{\partial\theta}\log p(\mathbf{w},\tau,\alpha,\mathcal{D}\,|M,\theta) \\ &= \frac{NN_o}{2}\log\frac{\tau}{2\pi} - E(\mathbf{w},\tau) \end{aligned} \tag{6.74}$$

where $q(\mathbf{w},\tau,\alpha\,|M,\mathcal{D},\theta)$ is the non normalized posterior for which the normalizing constant is exactly $p(\mathcal{D}\,|M)$.

The draws from the different parameters at the different temperatures are obtained in the same way as when sampling at unit temperature i.e. using Gibbs samples and Hybrid Monte Carlo but now at the different temperatures.

6.3.6 Robot arm prediction and selection of neural network complexity

The robot arm data was used by MacKay to show the performance of the evidence framework. Later Radford [16] used the data in a MCMC scenario similar to this, lately [5] the data was used also in a MCMC scenario to estimate $p(\mathcal{D}\,|M)$ in this paper a chaining approach was used.

The robot arm data are totally artificial. The data are generated by two function i.e. two outputs both taking the inputs x_1 and x_2. x_1 and x_2 represents some "joint angles" for which the robot arm positions in rectangular coordinates y_1 and y_2 as follows

$$y_1 = 2\cos x_1 + 1.3\cos(x_1+x_2) + \varepsilon_1 \tag{6.75}$$

$$y_2 = 2\sin x_1 + 1.3\sin(x_1+x_2) + \varepsilon_2 \tag{6.76}$$

where ε_1 and ε_2 is independent Gaussian noise with standard deviation 0.05. The angles are uniformly distributed x_1 in the intervals $\{-1.932; -0.453\} \wedge$

$\{0.453; 1.932\}$ and x_2 uniformly in $\{0.534; 3.142\}$. The interesting about this mapping is in theory it requires infinitely many neurons to implement when using a single layer feedforward network with tanh-activations. Even though, all kinds of Maximum Likelihood procedures will suggest using a finite number of hidden units when having finite data. This is of course due to over fitting. The evidence approach also suggests using a finite number of hidden units, also because some parameters have been fitted. The interesting question is does a sampling setup suggest using more hidden units and more interesting in this thesis is it consistent with our approximation of $p(\mathcal{D}\,|M)$. In [5] the results show favor of the hypothesis that the network complexity can grow with out over fitting.

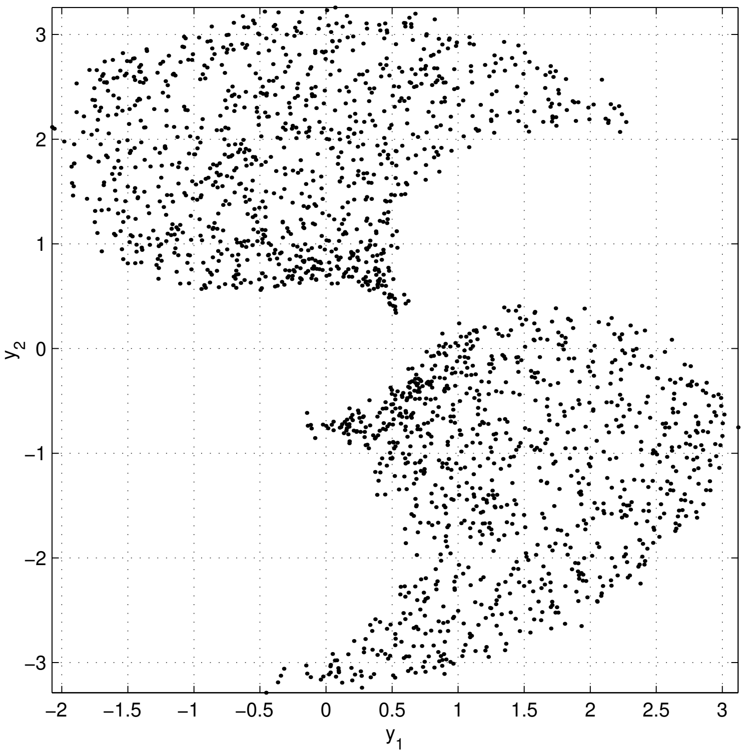


Figure 6.10: Robot arm output plotted against each other. The data are generated by drawing 2000 point from the input distribution. These are evaluated by the functions described in the text and added with independent Gaussian noise with standard deviation 0.05

On figure 6.3.6 can the robot arm output be seen in the rectangular coordinate system.

On figure 6.11 is the output plotted against the input this yields four com-

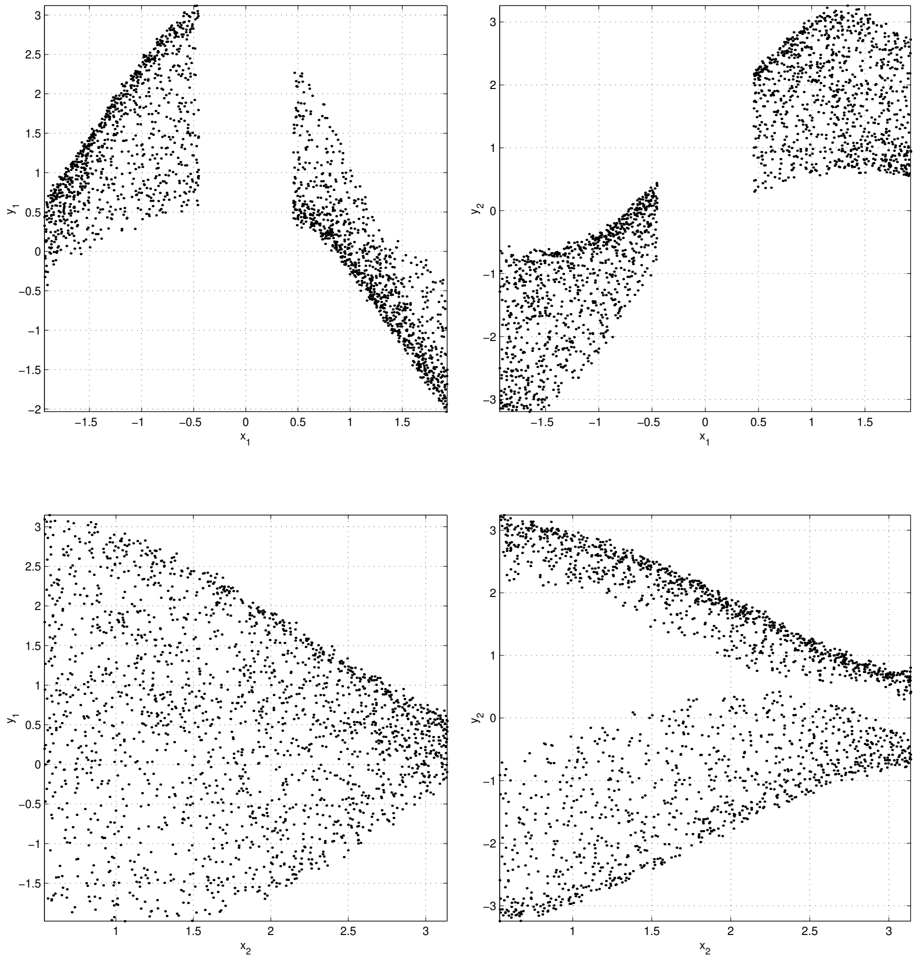


Figure 6.11: Robot arm data plotted against the inputs. The data are generated by drawing 2000 point from the input distribution. These are evaluated by the functions described in the text and added with independent Gaussian noise with standard deviation 0.05

binations when plotting in 2D.

In the following will two solutions to the robot arm prediction problem be described. Both solutions uses feedforward ANN's with a single hidden layer and tanh-activation. In both cases will the performance bee measured for network with 1 to 19 hidden units. The two approaches are a Maximum Likelihood based solution and the second a sampling based method. Unfortunately did I not have the time to use my Ensemble method.

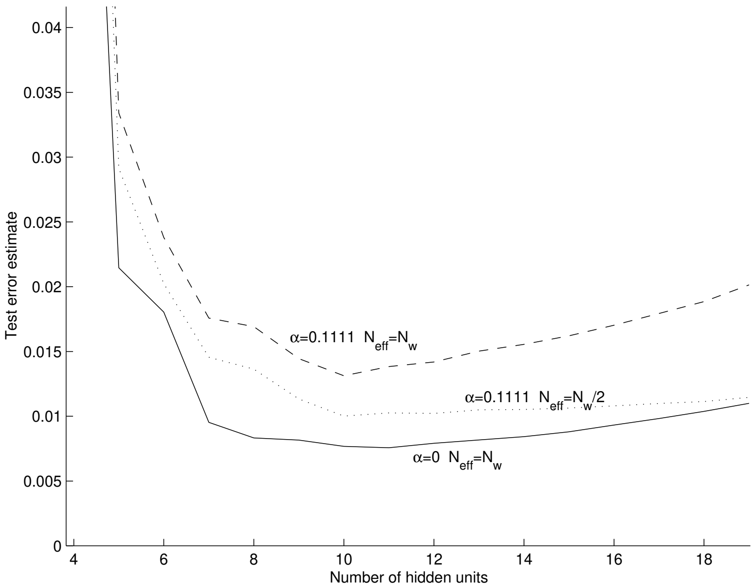


Figure 6.12: Lowest curve: the FPE-curve with zero weight decay. Upper curve the FPE-curve with a weight decay of 0.1111. Middle curve: the FPE-criteria with reduced number of parameters in the estimate precisely half the number of the number of parameters in the actual model

Maximum Likelihood Solution

The Maximum Likelihood solution is found by conjugate gradients. The optimization was run for 10 different seeds on a single data set containing 200 training examples. A single weight decay parameter was used to cover all the weights, this was set by hand to 10 different values linearly spread in the interval 0 to 1. The optimal weight decay and model was chosen by the FPE-criteria thus knowing this is only valid for no weight decay. The 10 different seeds seems to give rise to nearly the same solution, so these were averaged together and the decision was made upon this average.

When it comes to choosing between the weight decays using non weight decay has the lowest training performance. The FPE-criteria shows uniform over all the network sizes the same. Knowing that the FPE-criteria is not the right criteria to use, I tried to reduce the number of parameter in the FPE-criteria to half the number of parameters in the actual network to simulate an effective number of parameters. The result is shown on figure 6.12 the lowest curve shows the FPE with zero weight decay, the upper curve is the FPE with a weight decay of 0.1111 these curves uses in the FPE-criteria the actual number of parameters in the model, the middle curve is the FPE-criteria with reduced number of parameters in the estimate precisely half the number of the number of parameters in the actual model. We see that the FPE-curve for zero weight decay is uniformly the lowest justifying to choose zero weight decay. Just to mention, the choice is consistent with the test error, but if it was not it could not have altered the choice.

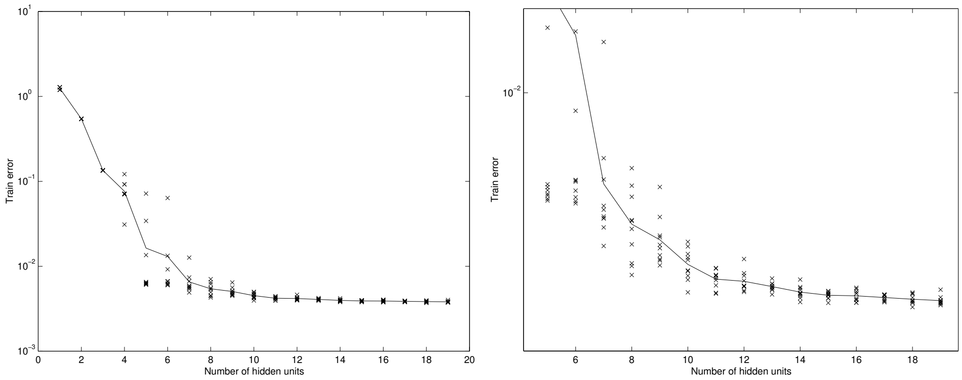


Figure 6.13: a: Training error for zero weight decay, the line shows the average training error and x shows the 10 different seeds. b: the same as a but zoomed. As seen the training error is declining for larger and larger models as expected.

When choosing amongst the models with different number of hidden units the zero weight decay is used and the FPE-curve is used as selection criteria, which is valid since the weight decay is zero. Figure 6.13 shows the training error when using zero weight decay. As expected the curve is a monotone declining function which can be due to a too small model or due to over fitting, if we chose the model based on the training error it self, the model with 19 hidden units would be chosen.

Figure shows the test error estimate by FPE. We see that the criteria has the lowest value when using 11 hidden units. As mentioned in the figure caption the choice is not definite due to the variance on the estimate, when taking this into account 8 or 12 hidden units could also be chosen.

We saw that the test error by the FPE-criteria suggested using 11 hidden units. On figure 6.15 is the true test error shown, made on 200 test examples. We see that there exists an "Occam valley" suggesting the use of 13 to 16 hidden units. This shows a two things:

- The ML over fits large models (In this setup no very significant)

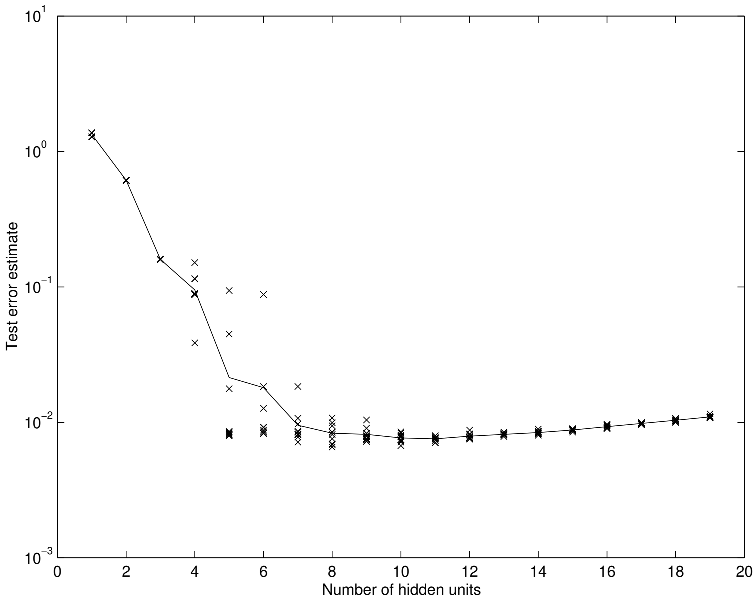


Figure 6.14: Test error estimate by FPE for zero weight decay, the line shows the average training error and x shows the 10 different seeds. As seen the test error estimate would suggest using 11 hidden units, but due to the variance it could as well be 8 or 12 hidden units.

- The FPE-criteria is pessimistic compared to the test error

The first is not very significant. The second problem can be explained by the fact that FPE is actually only valid if the model has capacity to implement the true function, so will FPE help to prevent over fit. If this is not the case we not only have a misfit in the parameters but also in the model it self. When having a wrong model we should allow using larger models than FPE suggests in the hope that a larger model is closer to the true model. So in this case where the true model can not be implemented by a finite network, which in general is the case, the GPE-criteria should be used, which can be less pessimistic towards larger models.

The mean test error [5] at 11 hidden units, the optimal model by FPE, was 0.00254 i.e. only slightly more than the true test error 0.00250. The model having the lowest training error, 19 hidden units, had a test error of 0.00256. The model having the lowest test error 13 or 16 hidden units yielded 0.00252.

[5] The test error on the figures are the squared error on both channels added together, divided by the number of inputs so the theoretic minimum is two times the true variance i.e. $2 \cdot 0.00250$. The test error in the following text is the test error from the figures divided by two so they are comparable with the true variance 0.00250

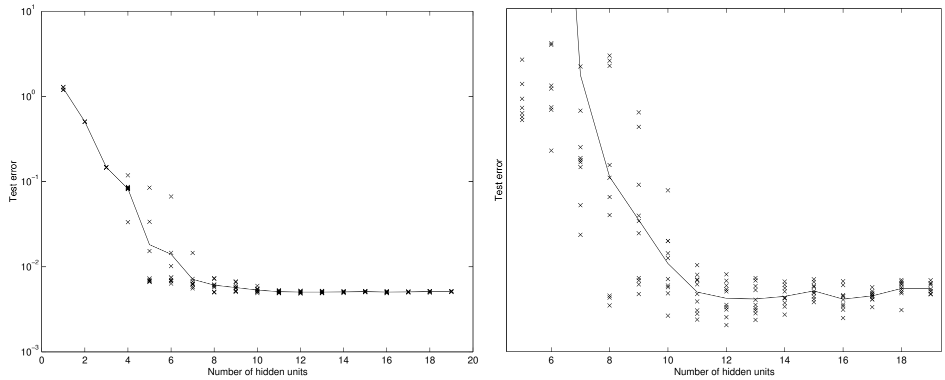


Figure 6.15: a: Test error for zero weight decay, the line shows the average training error and x shows the 10 different seeds. b: the same as a but zoomed. As seen the test error suggest using 13 hidden units, but due to the variance it could as well be 10 or 16 hidden units.

We see that the Maximum Likelihood method performs well on the robot arm data, but the phenomena expected is seen. When it comes to model selection the FPE-criteria selected a model that performed better than the largest model. But the choice is not consistent with the test error, which suggest a larger model. But the FPE-criteria is used wrongly as mentioned above, since we can not implement the true model with a finite number of hidden units. Instead the GPE-criteria should be used.

Monte Carlo solution to the Bayesian approach

In this section will the Bayesian framework be used on the robot arm data. The solution is provided by Monte Carlo estimates. When using Monte Carlo methods the estimate will not only suffer from the variance in the actual data, but also from variations in the posterior samples. Further more must one assure that the sampling method used, are capable of producing samples from the entire distribution in finite time. This can not be proven theoretically but we can hope that this can be fulfilled, and we can try to adjust the sampling mechanism to get as fast as possible around in the distribution. In this section will the evidence of the model be estimated, this will be compared to the test error.

In the setup have I used Gamma-priors

$$p(\tau) = \frac{w^s}{\Gamma(s)} \tau^{s-1} \exp -w\tau \mathbb{I}_{[0;\infty[}(\tau) \tag{6.77}$$

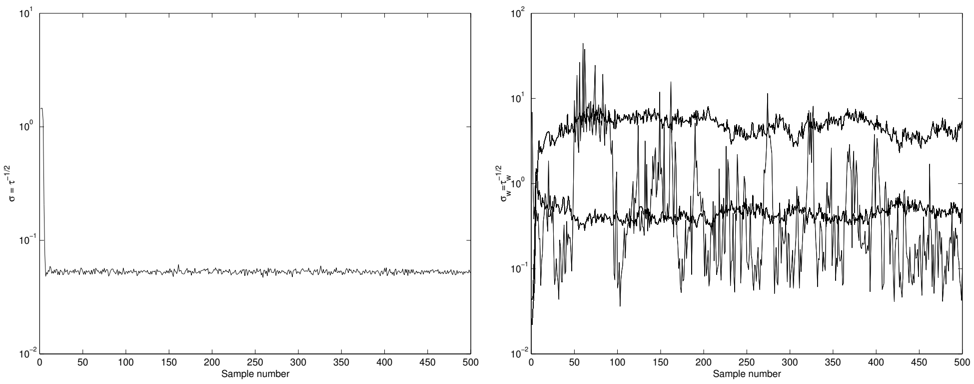


Figure 6.16: The figures shows $\tau^{-1/2}$ and ${\tau_w}^{-1/2}$ in the burn in phase for the largest model with 19 hidden units as they develop by time. Only every fourth sample of the 2000 are shown. a: shows the noise level which is found fairly fast. b: The three weight prior parameters, the most variate corresponds to the bias to output, the upper curve is the hidden to output weight group and the last the input and bias to hidden weights

on τ and on three weight groups $\tau_w^{(k)}$, $k = 1 \ldots 3$ the groups being bias to hidden weights and input to hidden weights, hidden to output weights and the last group bias to output. The parameters on the gamma noise prior was chosen vague $s_{tau} = 0.1$ and $w_{tau} = 0.1/100$ i.e. with a mean precision of 100 corresponding to $\sigma_{prior} = 0.1$. I will show that this is a very insensitive choice by not only using the standard robot arm data with a noise standard deviation of 0.05 but also by using a noise standard deviation of 0.5. The prior on the weights from bias to output is $s^{bo}_{tau_w} = 0.25$ and $w^{bo}_{tau_w} = .25/64$ and for the input to hidden and hidden biases $s^{ih}_{tau_w} = 0.25$ and $w^{ih}_{tau_w} = .25/64$ and lastly the hidden to output priors $s^{ho}_{tau_w} = 0.25$ and $w^{ho}_{tau_w} = .25/N_h$ i.e. the scale parameter scales with the number of hidden units assuring that when the number of hidden units goes to infinity the model will go towards a Gaussian process [16].

Some preliminary sampling setups was made to decide how the temperature distribution should be characterized by the constant k. It seemed that uniform chosen (inverse) temperatures yielded the best result in terms of Monte Carlo variance. 100 temperatures was drawn and sorted. The evidence is estimated for each of the 19 different models with 1 to 19 hidden units. This was performed both by starting at Bayes temperature $\theta = 1$ and at infinite temperature $\theta = 0$. When starting at Bayes temperature a "burn in phase" is required to reach the stationary distribution. The burn in was performed by 2000 Gibbs updates containing Hybrid Monte Carlo trajectories each consisting of 8000 leapfrog steps so in total 16 million leapfrog steps. Out of the 2000 states 100 states of the last 400 states was used for prediction. The leapfrog step size was chosen to yield an acceptance of around 80%.

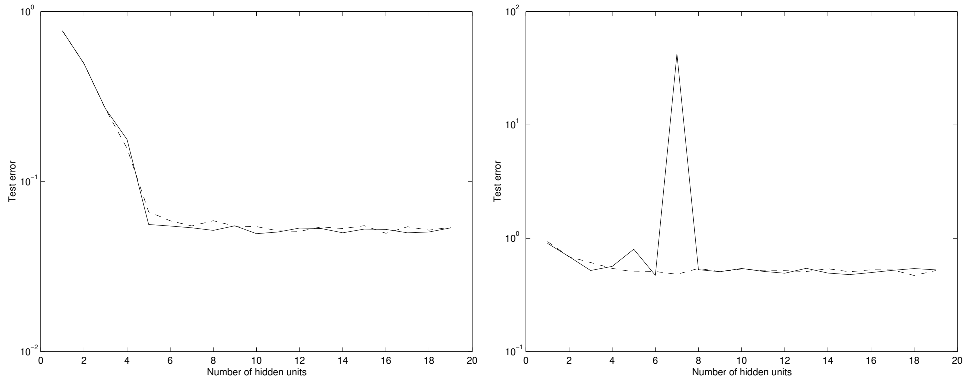


Figure 6.17: a: Test errors for the robot arm data with additive Gaussian noise with standard deviation 0.050. The dotted line correspond to the setup started at $\theta = 1$ and the other to the start at $\theta = 0$. b: The same as a. but with a noise standard deviation of 0.5, the data point falling "outside" the others are explained in the text.

On figure 6.16 is the evolution over time of the prior parameters shown. After 400 states marked on the figure as the 100^{th} state does it seem like the stationary distribution is reached, but the parameter for the bias to output group seem to fluctuate drastically up to the 800^{th} state. It is hard to justify if this fluctuation is a part of the stationary distribution or a convergence phenomena. The last hundred samples are used to represent the distribution i.e. every fourth of the last 400 states. This distance seems a little too short, but since they are used to estimate the mean network output it is still a central estimate, whereas the variance will be optimistic. For all the models less than 19 hidden units, the same burn in period was used. This ensures that we have reached the stationary distribution for all the models. In the setup where we start from $\theta = 0$ we start by traversing over the temperatures until reaching Bayes temperature where 400 Gibbs updates is taken with 8000 leapfrog steps in each, every 4^{th} of these updates are taken to represent the distribution.

On figure 6.17 is the test error shown for the two methods starting at inverse temperature $\theta = 0$ and $\theta = 1$ for the two noise levels 0.05 and 0.50. We see

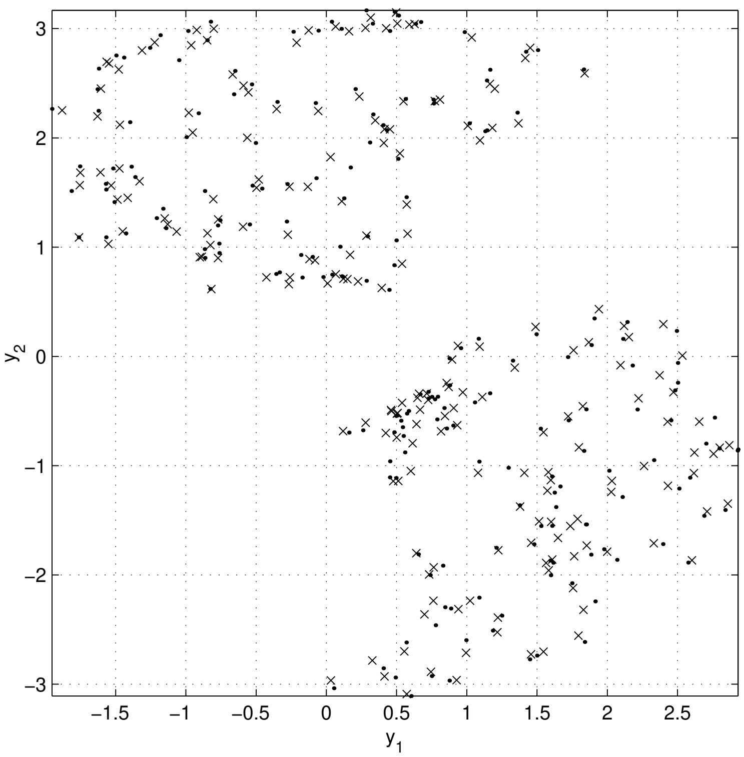


Figure 6.18: Predictions from the network with 19 hidden units. The crosses shows the targets and the dots marks the predictions. The data having 0.05 noise level. The simulation is started from $\theta = 1$ with the heavy burn in.

that the test error is still declining for larger models suggesting the use of the largest possible model. The result is not totally clear due to the variance, but no minimum occurs. The result is the same whether we are simulating from $\theta = 0$ or $\theta = 1$. The method of starting from $\theta = 0$ and simulate up to $\theta = 1$ can be faster than the heavy burn in phase at $\theta = 1$. This method used to reach the stationary distribution is called simulated annealing [15] and is a side offer when trying to estimate the evidence. The data point falling outside the others are explained later. This outlier data has to do with the temperature sweep. On figure 6.18 are the predictions shown together with the targets from the test set.

The temperature sweep is made by sampling succeeding temperatures. At each temperature is only 16 Gibbs updates made with corresponding Hybrid Monte Carlo updates for the weights. This is justified by the fact that we are at the stationary distribution when starting the temperature sweep from either $\theta = 0$ or $\theta = 1$ and then we are only altering the temperature a bit,

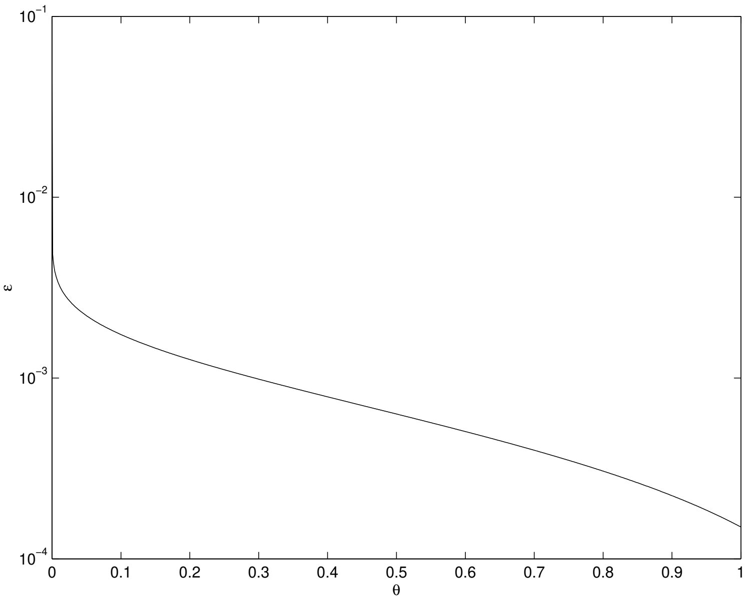


Figure 6.19: Step size in Hybrid Monte Carlo for different temperatures.

hoping that we are near the stationary distribution at this new temperature. Each Hybrid Monte Carlo update consists of 1000 leapfrog steps. The step length in the Hybrid Monte Carlo was made a function of the (inverse) temperature

$$\epsilon = 0.07 - \theta^{0.01}(0.07 - 0.00015); \tag{6.78}$$

see figure 6.19. This choice gave an acceptance of 80% to 95% at the different temperatures.

I will now turn back to the problem of estimating the evidence. The log evidence estimate is shown on figure 6.20. The figure shows four graphs, one for each of the starting temperatures and one for each of the noise levels. We see that the graphs in the same figure are different, which should not be the case, since the estimate should be independent of the starting point. What we see, is the estimate is biased towards the starting value. This can be verified by looking at the $U\left(\boldsymbol{\omega}_{\mathbf{i}}, \boldsymbol{\theta}_{\mathbf{i}}\right)$-function the average of this function is the evidence estimate.

On figure 6.21 is the $U\left(\boldsymbol{\omega}_{\mathbf{i}}, \theta_i\right)$-function shown for the case of 18 hidden units and a noise level of 0.05. We see that the $U\left(\boldsymbol{\omega}_{\mathbf{i}}, \boldsymbol{\theta}_{\mathbf{i}}\right)$-function "latches" into the most probable model at the starting temperature. The reason for this latching is of course due to the sampling method. By remembering the case in the linear model with the bimodal joint distribution of the two

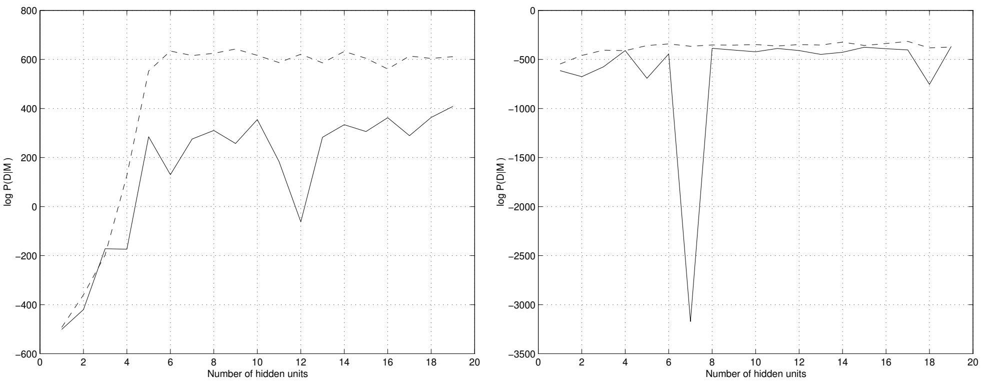


Figure 6.20: a: Log evidence estimate of the different models when the noise level is 0.05 . The dotted line corresponds to start at $\theta = 1$ and the other to start at $\theta = 0$. b: The same as a. but with a noise level of 0.5

hyper parameters τ and τ_w we saw that it could be very difficult to "move" τ_w, this only occurring when a majority of the weights occasionally move towards the other level. The same is going on in this setup, hence when starting from either of the sides we latches into the model most likely at this temperature i.e. we latches into one of the modes in the posterior. This is crucial for the estimate of the log evidence, which is then too large or too small depending on which model that are latched onto. This can of course be solved by sampling for a longer duration between each sample, this may not bee the best method. Another method is to sample from a random place at each temperature requiring a burn in phase at each temperature, this will also be very time consuming. Yet another approach is to make a sampling procedure that is capable of suggesting samples from the other model than the current, and produce a sample in the normal way, and then make a procedure to choose amongst these two samples. The samples can be produced by starting from both temperatures. This is of course an ad hoc procedure which is difficult to generalize.

The outlier points in the test error and the evidence will now be addressed. It is clear that the test error, for the model with 7 hidden units estimated from $\theta = 0$ on the data set with noise standard deviation 0.50, is wrong. This happens because the true model is newer found, which is due to very bad initial start under the prior at $\theta = 0$. This is actually also the reason why the model with 12 hidden units shows a low evidence on figure 6.20.a. That the test error is not that bad for this model is because the "true" model is found before the predictions occurs. The phenomena can be seen on figure

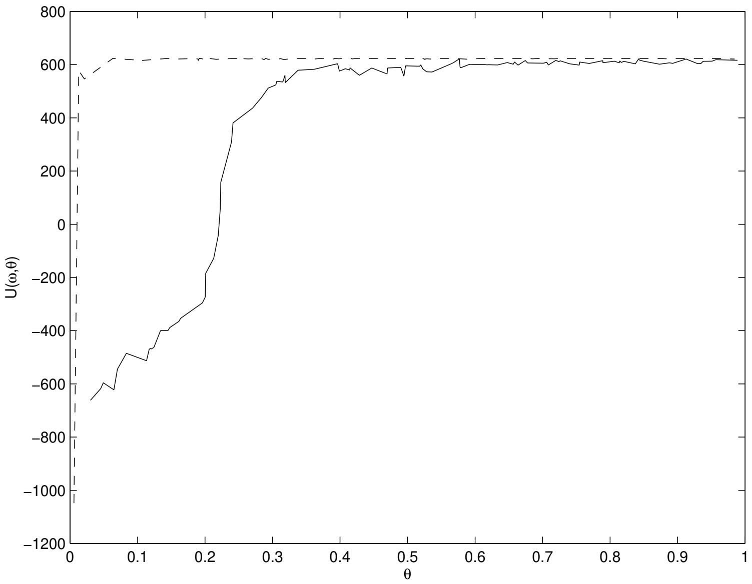


Figure 6.21: The $U(\boldsymbol{\omega}_\mathbf{i}, \boldsymbol{\theta}_\mathbf{i})$-function for the model with 18 hidden units and a noise level of 0.05. The dotted line is the result from starting in $\theta = 1$ the other in $\theta = 0$.

6.22. As explained in the figure caption is three levels seen in the $U(\boldsymbol{\omega}_\mathbf{i}, \theta_i)$-function. The upper level corresponds to a model where the true noise level is found, the middle level to a model where all the signal is modeled as noise and the lower level to a model with higher noise level than the total signal level. This can be verified by looking at figure 6.22.b. The 100 samples from 101 to 200 are the noise levels drawn in the temperature sweep, where the three levels also are found. The two levels corresponding to the "true" model and the model believing all the signal to bee noise are found in all the setups. The third level having an enormous noise standard deviation only arises because of the very vague prior. On figure 6.22.b is the first 100 samples of the noise level sampled from the prior i.e. $\theta = 0$ when we starts to see data, the 101 sample, the state is in a very unlikely position when given some data. But since the noise level is so high is it also probable to pick very large network outputs and hence weights, since the likelihood is insensitive to this. This makes a kind of latching procedure into this very unlikely state. At higher θ the likelihood is seeing more examples and this odd noise level becomes less probable. Fortunately the model at the next level is found, and at last is the true model found giving the reason why the test error is not obscured. Because the true noise level is found is the evidence estimate not that extreme as for the model with 7 hidden units. When examining this

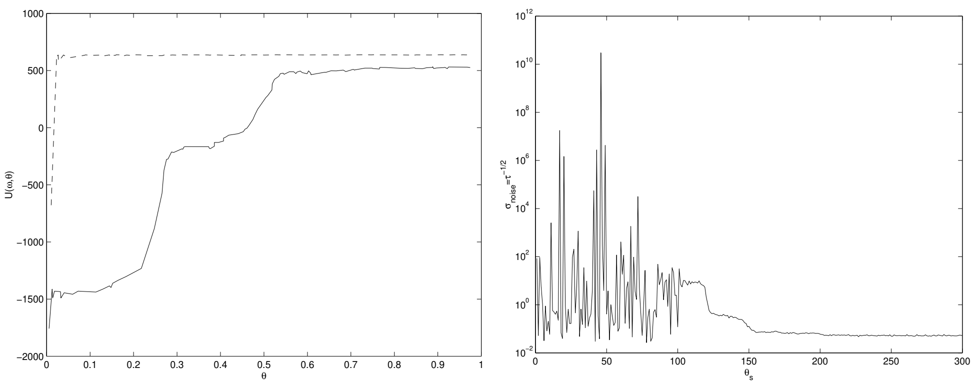


Figure 6.22: a: The hysteresis effect for the model with 12 hidden units on the data with noise standard deviation 0.05. The full drawn curve is the method started at $\theta = 0$. On this curve is three levels seen, for an explanation see the text. b: estimated noise level for the same model as on figure a. The first hundred samples is from the prior, the next hundred from the (inverse) temperature sweep from $\theta = 0$ to $\theta = 1$ and the last hundred samples is from the prediction phase i.e. hundred samples at Bayes temperature $\theta = 1$.

model one encounters the same problem but worse, since the true noise level is newer found.

6.3.7 Comparison of the Maximum Likelihood approach and the Monte Carlo estimation

Summarizing shortly the two methods performance. Since an finite ANN seldom is the "true" process describing data an infinite network should be used. In a Maximum Likelihood setup will large ANN's tend to bee over fitted. The FPE test error estimate is though too pessimistic when trying to select the best sized ANN, due to the fact that the model is newer the "true" one. Instead GPE have to be used. The test error of the ANN in the Maximum Likelihood suggested the use of a finite ANN but though not significant.

In the Bayesian setup does the test error fall for still larger models. This by it self justifies the use of as large as possible a model. The evidence estimation is very difficult, for this reason is it not consistent with the test error. Estimating the evidence for different sized ANN's is awkward, since we know that the largest possible model is the best one when the data is not coming from a finite ANN. Comparing different models in a Bayesian context only makes sense when there exists some kind of hypothesis to be tested, the model selection to reduce complexity is nonsense in a Bayesian setup since no over fitting occurs.

Comparing the test performance of the ML ANN and the Bayes ANN does in the case of the robot arm data not suggest one or the other. I think this is due to the relative large training data set making the Maximum Likelihood estimate good. Furthermore is the robot arm functions described well by 12 or so hidden units, so the benefit from using more hidden units is small compared to the estimation variance in larger networks.

7. Conclusion

The Bayesian model concept has been examined already, in the second chapter did we see an example of why Bayes performs that well, it introduces a nice and efficient way to introduce bias via the priors. When Maximum Likelihood is used, a lot of effort is used to regularize the model, which corresponds to introduction of bias. The two are nearly the same priors or regularization. I think a major difference is that Bayesians afterwards average over all the possible parameter values weighted according to the posterior probability whereas Maximum Likelihood tries to find a single optimal point estimate of the parameters. This can lead to overfitting but so can Bayes if a too wrong prior is used. The overfitting of the Maximum Likelihood can be removed by having the right regularizer, arguing for that both methods can be optimal. Regularization in a Bayesian setup is more probabilistic than in Maximum Likelihood. The averaging in Bayesian learning is a major part of the regularization, which can be difficult or even impossible to formulate into a parameterized regularizer in the Maximum Likelihood setup.

When talking decision theory both Bayes and Maximum Likelihood are methods to solve the problem of interest. All kinds of decision problems should start out with specifying the model, the precise competing hypotheses and the costs and losses of the possible decisions and their outcomes. After these preliminary things a strategy should be made like minimizing the worst loss possible or minimizing the average cost. Of course either of these two methods only put a single constraint on the decision making. Now Bayesian or Maximum Likelihood methods can be taken into account. Yielding a modeling procedure of the different hypotheses. The one model having the lowest min/max, average or what ever cost is the hypothesis chosen. This leads back to the modelling procedure it self and hence the same problems: priors and regularization.

All the discussion which method being the best is hard even in theory. So being more pragmatic one must use the methods on the problem of interest.

For that reason are the methods used to solve regression problems. Two models have been examined on two problems the linear model and the Artificial Neural Network (ANN). These models often suffer from over fitting due to the large amount of parameters. This has been a motivation in it self to select amongst models. The conclusion is, based on the independent test errors, that this is not a motivation in Bayesian modelling since this does not suffer from over fitting, this was the case both for the linear model and the ANN model. Bayesian model selection it self can be used in hypotheses test.

Two different hypotheses was setup. Selection of the number of input to a linear model and selection of the number of hidden units in an ANN. For the linear model analytic Bayesian model selection proves superior to Maximum Likelihood. When comparing the results based on the selection criterion and test error the Bayesian method also proves consistency, where as Maximum Likelihood needs penaliziation to yield consistency, but having the right asymptotic penalty Maximum Likelihood also selects the true one. Using the sampling approximation in Bayes decision express the same result as for the analytic case, but the result is not clear.

In the ANN setup both Maximum Likelihood and Bayes performs rather good when comparing the test errors to the theoretic minimum. The Bayesian test error suggests using as large as possible a model, whereas the Maximum Likelihood test error suggests using a model with a finite number of hidden units. So if a hypothesis was made like: how many hidden units should be used to describe the input output relation of interest, based on the test error, the Bayesian method would yield the true answer in the case of the robot arm data since in theory infinitely many should be used. The Maximum Likelihood method would give a wrong answer all due to over fitting, but the wrong answer is not the same as we not have found a sufficiently accurate model. This shows the benefit of using Bayes in decision setups. But the hope was that the same decision could be made without setting data aside to validate/test the model. This can be carried out either by using the training error or by using the Likelihood of the training data in the Maximum Likelihood setup or in the Bayesian setup the evidence of the model. By using the training error Maximum Likelihood will in general select a too large model due to over fitting but Bayes would yield a training error consistent with the test error and hence a more accurate decision is made. The Maximum Likelihood selection criteria again suffers from over fitting but is easy to calculate. The evidence of the model yields good decisions if it can be derived, when it have to be estimated as was the case in the ANN setup the result is unclear.

The method used, path sampling, has the potential to yield the right Bayes answer. But sampling in the ANN setup makes the estimation very time consuming. The idea of ordering the temperatures to reduce the computation time, was maybe not the best to do, though at least as good as not doing it. The method can be improved, not only by sampling for a longer duration, but by constructing a sampling method that are even better to move around in the distribution of interest. But even though path sampling did not yield consistent answers, a lot of interesting phenomena was discovered. The evidence approximation is calculated as the ratio between two models, the temperature sweep so to say selects amongst the models and at a certain temperature a rapid phase transition occurs. This phase transition is actually what makes the estimation hard, since in the temperature interval where the phase transition can occur, different modes exists in the joint distribution of the hyper parameters. The major challenge is then to construct a sampling method that easily can propose samples from the competing modes. There is a hope that this is possible since we can identify two modes by starting at the other end of the temperatures. But how this should be used is not clear.

Yet another approach to estimate the evidence has been proposed. By ensemble learning an approximation to the posterior can be found. An ensemble method for the ANN has been derived. By using this ensemble estimate as the sampling distribution the normalizing constant of the posterior, which is the same as the evidence, can be estimated by importance sampling.

Appendix A. Analytic derivation for the linear model

A.1 Predictive distribution: Linear model

The following derivations, are directly related to the results in 6.2

If we assume the model to be linear, with a single outputs, then we will have

$$\mathbf{f}(\mathbf{x};\mathbf{w}) = \mathbf{x}^T\mathbf{w} \tag{A.1}$$

By using a Gaussian noise model, the likelihood becomes

$$f(\mathbf{Y}\,|\mathbf{X},\boldsymbol{\omega},\mathbf{f}) = \left|\frac{\tau}{2\pi}\right|^{\frac{N}{2}} \exp\left(-\frac{\tau}{2}\left(\mathbf{X}^T\mathbf{w}-\mathbf{Y}\right)^T\left(\mathbf{X}^T\mathbf{w}-\mathbf{Y}\right)\right) \tag{A.2}$$

where τ is the noise precision, and $\boldsymbol{\omega}^T = \left[\mathbf{w}^T, \tau\right]$. We use a Gaussian prior on the weights $\mathbf{w}$ with a precision $\tau_w = \rho\tau$ and a mean vector equal to the null vector. The noise precision is given a gamma prior with parameters α and β. This yields

$$\begin{aligned}
\pi(\mathbf{w}\,|\rho,\tau) &= \left|\frac{\rho\tau}{2\pi}\right|^{\frac{N_w}{2}} \exp\left(-\frac{\rho\tau}{2}\mathbf{w}^T\mathbf{w}\right) \\
\pi(\tau\,|\alpha,\beta) &= \frac{1}{\Gamma(\alpha)\,\beta^{\alpha}}\tau^{\alpha-1}\exp-\frac{\tau}{\beta}
\end{aligned}$$

where N_w is the number of weights. The total prior then becomes

$$\pi(\boldsymbol{\omega}\,|\rho,\alpha,\beta) = \left|\frac{\rho\tau}{2\pi}\right|^{\frac{N_w}{2}} \frac{1}{\Gamma(\alpha)\,\beta^{\alpha}}\tau^{\alpha-1}\exp-\tau\left(\frac{\rho}{2}\mathbf{w}^T\mathbf{w}+\frac{1}{\beta}\right) \tag{A.3}$$

which is in the family of normal-gamma distributions. The posterior is then composed by the prior and the likelihood

$$\begin{aligned}
k(\boldsymbol{\omega}\,|\mathcal{D},\rho,\alpha,\beta) &= \frac{(a/2)^{\alpha+\frac{N}{2}-\frac{N_w}{2}}}{\boldsymbol{\Gamma}\left(\alpha+\frac{N}{2}-\frac{N_w}{2}\right)|\boldsymbol{\Sigma}|^{1/2}(2\pi)^{\frac{N_w}{2}}}\tau^{\alpha+\frac{N+N_w}{2}-1} \\
&\qquad \exp-\frac{\tau}{2}\left(\left(\mathbf{w}-\mathbf{w}_0\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{w}-\mathbf{w}_0\right)+a\right) \\
\boldsymbol{\Sigma}^{-1} &= \mathbf{X}\mathbf{X}^T+\rho\mathbb{I} \\
\mathbf{w}_0 &= \boldsymbol{\Sigma}\mathbf{X}\mathbf{Y} \\
a &= \frac{2}{\beta}-\mathbf{w}_0^T\boldsymbol{\Sigma}^{-1}\mathbf{w}_0+\mathbf{Y}^T\mathbf{Y}
\end{aligned} \tag{A.4}$$

Now the predictive distribution can be written

$$p\left(y^{(N+1)}\left|\mathbf{x}^{(N+1)},\mathcal{D},\rho,\alpha,\beta,\mathbf{f}\right.\right)=\int f\left(y^{(N+1)}\left|\mathbf{x}^{(N+1)},\boldsymbol{\omega}',\mathbf{f}\right.\right)k\left(\boldsymbol{\omega}'\left|\mathcal{D},\rho,\alpha,\beta\right.\right)d\boldsymbol{\omega}' \tag{A.5}$$

so

$$p\left(y^{(N+1)}\left|\mathbf{x}^{(N+1)},\mathcal{D},\rho,\alpha,\beta,\mathbf{f}\right.\right)\propto \iint \tau^{\alpha+\frac{N+N_w+1}{2}-1}\exp -E\left(y^{(N+1)},\mathbf{w},\tau\right)d\mathbf{w}d\tau \tag{A.6}$$

$$\begin{aligned}E\left(y^{(N+1)},\mathbf{w},\tau\right)&=\frac{\tau}{2}\left(\left(\mathbf{w}-\mathbf{w}_0\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{w}-\mathbf{w}_0\right)+a\right)\\&+\frac{\tau}{2}\left(\mathbf{w}^T\mathbf{x}^{(N+1)}-y^{(N+1)}\right)^T\left(\mathbf{w}^T\mathbf{x}^{(N+1)}-y^{(N+1)}\right)\end{aligned}$$

We can rewrite the integrals

$$\begin{aligned}p&\propto\int\tau^{\alpha+\frac{N+1}{2}-1}\exp-\frac{\tau}{2}b\int\exp-\frac{\tau}{2}\left(\mathbf{w}-\overline{\mathbf{w}}\right)^T\overline{\boldsymbol{\Sigma}}^{-1}\left(\mathbf{w}-\overline{\mathbf{w}}\right)d\mathbf{w}d\tau\\\overline{\boldsymbol{\Sigma}}^{-1}&=\mathbf{X}\mathbf{X}^T+\mathbf{x}^{(N+1)}\mathbf{x}^{(N+1)^T}+\rho\mathbb{I}=\boldsymbol{\Sigma}^{-1}+\mathbf{x}^{(N+1)}\mathbf{x}^{(N+1)^T}\\\overline{\mathbf{w}}&=\overline{\boldsymbol{\Sigma}}\left(\mathbf{X}\mathbf{Y}+\mathbf{x}^{(N+1)}y^{(N+1)}\right)\\b&=\frac{2}{\beta}-\overline{\mathbf{w}}^T\overline{\boldsymbol{\Sigma}}^{-1}\overline{\mathbf{w}}+\mathbf{Y}^T\mathbf{Y}+y^{(N+1)^T}y^{(N+1)}\end{aligned}\tag{A.7}$$

The integral over $\mathbf{w}$ is a Gaussian integral yielding

$$\begin{aligned}p\left(y^{(N+1)}\left|\mathbf{x}^{(N+1)},\mathcal{D},\rho,\alpha,\beta,\mathbf{f}\right.\right)&\propto\int\tau^{\alpha+\frac{N+N_w+1}{2}-1}\exp-\frac{\tau}{2}b\frac{\left|\overline{\boldsymbol{\Sigma}}\right|^{1/2}(2\pi)^{\frac{N_w}{2}}}{\tau^{\frac{N_w}{2}}}d\tau\\&=\left|\overline{\boldsymbol{\Sigma}}\right|^{1/2}(2\pi)^{\frac{N_w}{2}}\int\tau^{\alpha+\frac{N+1}{2}-1}\exp-\frac{\tau}{2}bd\tau\end{aligned}\tag{A.8}$$

This is simply a gamma integral, so

$$p\left(y^{(N+1)}\left|\mathbf{x}^{(N+1)},\mathcal{D},\rho,\alpha,\beta,\mathbf{f}\right.\right)\propto\left|\overline{\boldsymbol{\Sigma}}\right|^{1/2}(2\pi)^{\frac{N_w}{2}}\Gamma\left(\alpha+\frac{N+1}{2}\right)\left(\frac{2}{b}\right)^{\alpha+\frac{N+1}{2}} \tag{A.9}$$

We can see that

$$p\left(y^{(n+1)}\left|\mathbf{x}^{(n+1)},\mathcal{D},\rho,\alpha,\beta,\mathbf{f}\right.\right)\propto \left(\frac{2}{\beta}-\left(\mathbf{X}\mathbf{Y}+\mathbf{x}^{(n+1)}y^{(n+1)}\right)^T\overline{\boldsymbol{\Sigma}}\left(\mathbf{X}\mathbf{Y}+\mathbf{x}^{(n+1)}y^{(n+1)}\right)+\mathbf{Y}^T\mathbf{Y}+y^{(n+1)}y^{(n+1)}\right)^{-\frac{2\alpha+N+1}{2}} \tag{A.10}$$

this can be rewritten into

$$p\left(y^{(n+1)}\left|\mathbf{x}^{(n+1)}, \mathcal{D}, \rho, \alpha, \beta, \mathbf{f}\right.\right) \propto \tag{A.11}$$
$$\left(\frac{2}{\beta}-\frac{\overline{y}^2}{\kappa}-\mathbf{Y}^T\mathbf{X}^T\overline{\mathbf{\Sigma}}\mathbf{X}\mathbf{Y}+\mathbf{Y^T}\mathbf{Y}+\frac{1}{\kappa}\left(y^{(n+1)}-\overline{y}\right)^2\right)^{-\frac{2\alpha+N+1}{2}}$$

where

$$\begin{aligned} \overline{y} &= \kappa\mathbf{x}^{(n+1)^T}\overline{\mathbf{\Sigma}}\mathbf{X}\mathbf{Y} \\ \frac{1}{\kappa} &= 1-\mathbf{x}^{(n+1)^T}\overline{\mathbf{\Sigma}}\mathbf{x}^{(n+1)} \end{aligned}$$

The predictive distribution can be rewritten into

$$p\left(y^{(n+1)}\left|\mathbf{x}^{(n+1)}, \mathcal{D}, \rho, \alpha, \beta, \mathbf{f}\right.\right) \propto \tag{A.12}$$
$$\left(1+\frac{2\alpha+N}{\kappa\left(\frac{2}{\beta}-\frac{\overline{y}^2}{\kappa}-\mathbf{Y}^T\mathbf{X}^T\overline{\mathbf{\Sigma}}\mathbf{X}\mathbf{Y}+\mathbf{Y}^T\mathbf{Y}\right)}\frac{\left(y^{(n+1)}-\overline{y}\right)^2}{2\alpha+N}\right)^{-\frac{2\alpha+N+1}{2}}$$

Which can be normalized since $y^{(n+1)} \sim t\left(2\alpha+N, \overline{y}, \frac{\kappa\frac{2}{\beta}-\overline{y}^2-\kappa\mathbf{Y}^T\mathbf{X}^T\overline{\mathbf{\Sigma}}\mathbf{X}\mathbf{Y}+\kappa\mathbf{Y}^T\mathbf{Y}}{2\alpha+N}\right)$
hence

$$p\left(y^{(n+1)}\left|\mathbf{x}^{(n+1)}, \mathcal{D}, \rho, \alpha, \beta, \mathbf{f}\right.\right) = \tag{A.13}$$
$$\frac{\Gamma\left(\frac{2\alpha+N+1}{2}\right)/\Gamma\left(\frac{2\alpha+N}{2}\right)}{\sqrt{\pi\left(\kappa\frac{2}{\beta}-\overline{y}^2-\kappa\mathbf{Y}^T\mathbf{X}^T\overline{\mathbf{\Sigma}}\mathbf{X}\mathbf{Y}+\kappa\mathbf{Y}^T\mathbf{Y}\right)}}\left(1+\frac{\left(y^{(n+1)}-\overline{y}\right)^2}{\kappa\left(\frac{2}{\beta}-\mathbf{Y}^T\mathbf{X}^T\overline{\mathbf{\Sigma}}\mathbf{X}\mathbf{Y}+\mathbf{Y}^T\mathbf{Y}\right)-\overline{y}^2}\right)^{-\frac{2\alpha+N+1}{2}}$$

The evidence of the model can also be calculated since

$$k\left(\boldsymbol{\omega}\left|\mathcal{D}, \rho, \alpha, \beta\right.\right) = \frac{\pi\left(\boldsymbol{\omega}\left|\rho, \alpha, \beta\right.\right) f\left(\mathbf{Y}\left|\mathbf{X}, \boldsymbol{\omega}, \mathbf{f}\right.\right)}{\int \pi\left(\boldsymbol{\omega}'\left|\rho, \alpha, \beta\right.\right) f\left(\mathbf{Y}\left|\mathbf{X}, \boldsymbol{\omega}', \mathbf{f}\right.\right) d\boldsymbol{\omega}'} \tag{A.14}$$

where the denominator is the evidence of the model

$$f\left(\mathbf{Y}\left|\mathbf{X}, \rho, \alpha, \beta, \mathbf{f}\right.\right) = \int \pi\left(\boldsymbol{\omega}'\left|\rho, \alpha, \beta\right.\right) f\left(\mathbf{Y}\left|\mathbf{X}, \boldsymbol{\omega}', \mathbf{f}\right.\right) d\boldsymbol{\omega}' \tag{A.15}$$

It can easily be derived

$$f\left(\mathbf{Y}\left|\mathbf{X}, \rho, \alpha, \beta, \mathbf{f}\right.\right) = \frac{\Gamma\left(\frac{2\alpha+N}{2}\right)/\Gamma\left(\alpha\right)\left|\mathbb{I}-\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}\right|^{\frac{1}{2}}}{\left(2\pi\beta^{-1}\right)^{\frac{N}{2}}}\left(1+\frac{\mathbf{Y}^T\left(\mathbb{I}-\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}\right)\mathbf{Y}}{2\beta^{-1}}\right)^{-\frac{2\alpha+N}{2}} \tag{A.16}$$

i.e. $\mathbf{Y}$ is t distributed

$$\mathbf{Y} \sim t\left(2\alpha, \mathbf{0}, 2\alpha\frac{\mathbb{I}-\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}}{2\beta^{-1}}\right) \tag{A.17}$$

Appendix B. Conditional distributions of the linear model

The conditional distributions in the linear model with independence between τ and τ_w and inverse temperature θ can be written

$$
\begin{aligned}
p\left(\mathbf{w}\left|\mathbf{Y},\mathbf{X},\tau,\tau_w,\theta,\mathbf{f}\right.\right) &= \frac{p\left(\mathbf{w}\left|\tau_w\right.\right) f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)^{\theta}}{\int p\left(\mathbf{w}'\left|\tau_w\right.\right) f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w}',\tau,\mathbf{f}\right.\right)^{\theta} d\mathbf{w}'} && \text{(B.1)}\\
p\left(\tau\left|\mathbf{Y},\mathbf{X},\mathbf{w},\tau_w,\theta,\mathbf{f}\right.\right) &= \frac{\pi\left(\tau\right) f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)^{\theta}}{\int \pi\left(\tau'\right) f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau',\mathbf{f}\right.\right)^{\theta} d\tau'} && \text{(B.2)}\\
p\left(\tau_w\left|\mathbf{w}\right.\right) &= \frac{\pi\left(\tau_w\right) p\left(\mathbf{w}\left|\tau_w\right.\right)}{\int \pi\left(\tau'_w\right) p\left(\mathbf{w}\left|\tau'_w\right.\right) d\tau'_w} && \text{(B.3)}
\end{aligned}
$$

The result from doing this is now to be derived. First we find

$$
\begin{aligned}
&p\left(\mathbf{w}\left|\mathbf{Y},\mathbf{X},\tau,\tau_w,\mathbf{f}\right.\right) \propto p\left(\mathbf{w}\left|\tau_w\right.\right) f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)^{\theta}\\
&\propto \exp -\frac{\tau_w}{2}\mathbf{w}^T\mathbf{w} - \frac{\theta\tau}{2}\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}\right)^T\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}\right) \qquad \text{(B.4)}
\end{aligned}
$$

which can be rewritten into

$$
p\left(\mathbf{w}\left|\mathbf{Y},\mathbf{X},\tau,\tau_w,\mathbf{f}\right.\right) \propto \exp -\frac{1}{2}\mathbf{w}^T\left(\theta\tau\mathbf{X}\mathbf{X}^T+\tau_w\mathbb{I}\right)\mathbf{w} + \theta\tau\mathbf{Y}^T\mathbf{X}^T\mathbf{w} \qquad \text{(B.5)}
$$

which can be normalized since the distribution is Gaussian

$$
\begin{aligned}
p\left(\mathbf{w}\left|\mathbf{Y},\mathbf{X},\tau,\tau_w,\mathbf{f}\right.\right) &= \frac{1}{\sqrt{\left|2\pi\mathbf{\Sigma}_w^{-1}\right|}}\exp -\frac{1}{2}\left(\mathbf{w}-\mathbf{w}_0\right)^T\mathbf{\Sigma}_w^{-1}\left(\mathbf{w}-\mathbf{w}_0\right)\\
\mathbf{\Sigma}_w^{-1} &= \left(\theta\tau\mathbf{X}\mathbf{X}^T+\tau_w\mathbb{I}\right)\mathbf{w}_0^T\\
&= \theta\tau\mathbf{Y}^T\mathbf{X}^T\mathbf{\Sigma}_w && \text{(B.6)}
\end{aligned}
$$

hence Gaussian.

The noise precision can be found by

$$
\begin{aligned}
&p\left(\tau\left|\mathbf{Y},\mathbf{X},\mathbf{w},\tau_w,\theta,\mathbf{f}\right.\right) \propto \pi\left(\tau\right) f\left(\mathbf{Y}\left|\mathbf{X},\mathbf{w},\tau,\mathbf{f}\right.\right)^{\theta}\\
&\propto \tau^{\alpha-1+\theta N/2}\exp -\tau\left(\beta^{-1}+\frac{\theta}{2}\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}\right)^T\left(\mathbf{Y}-\mathbf{X}^T\mathbf{w}\right)\right) \qquad \text{(B.7)}
\end{aligned}
$$

where N is the number of examples. As seen this is a Gamma distribution so the normalized version is

$$
\begin{aligned}
p\left(\tau\left|\mathbf{Y},\mathbf{X},\mathbf{w},\theta,\mathbf{f}\right.\right) &= \frac{\left(b^{-1}\right)^{a}}{\Gamma\left(a\right)}\tau^{a-1}\exp-\tau b^{-1} \\
a &= \alpha+\theta N/2 \\
b^{-1} &= \beta^{-1}+\frac{\theta}{2}\left(\mathbf{Y}-\mathbf{X}^{T}\mathbf{w}\right)^{T}\left(\mathbf{Y}-\mathbf{X}^{T}\mathbf{w}\right)
\end{aligned}
\tag{B.8}
$$

For the precision on the weights it is easy to derive

$$
\begin{aligned}
p\left(\tau_{w}\left|\mathbf{w}\right.\right) &\propto \pi\left(\tau_{w}\right)p\left(\mathbf{w}\left|\tau_{w}\right.\right) \\
&\propto \tau_{w}^{\alpha_{w}-1+N_{i}/2}\exp-\tau_{w}\left(\beta_{w}^{-1}+\frac{1}{2}\mathbf{w}^{T}\mathbf{w}\right)
\end{aligned}
\tag{B.9}
$$

where N_i is both the number of weights and number of inputs. As seen this is again a Gamma distribution

$$
\begin{aligned}
p\left(\tau_{w}\left|\mathbf{w}\right.\right) &= \frac{\left(b_{w}^{-1}\right)^{a_{w}}}{\Gamma\left(a_{w}\right)}\tau_{w}^{a_{w}-1}\exp-\tau_{w}b_{w}^{-1} \\
a_{w} &= \alpha_{w}-N_{i}/2 \\
b_{w}^{-1} &= \beta_{w}^{-1}+\frac{1}{2}\mathbf{w}^{T}\mathbf{w}
\end{aligned}
\tag{B.10}
$$

Appendix C. Ensemble estimate of the ANN

C.1 Approximating the ANN

The ANN-model are described by the parameters $\mathbf{w}$, τ and $\boldsymbol{\alpha}$ i.e. the network weights, the noise precision and the weight prior precision, the gamma distribution hyper parameters are fixed at specific values. The true posterior at a specific temperature

$$p\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)\propto p\left(\boldsymbol{\alpha}\right)p\left(\tau\right)p\left(\mathbf{w}\left|\boldsymbol{\alpha}\right.\right)p^{\theta}\left(\left\{y_i\right\}_{i=1}^{N}\middle|\left\{\mathbf{x}_i\right\}_{i=1}^{N},\tau,\mathbf{w}\right) \tag{C.1}$$

where $\boldsymbol{\alpha}=\{\alpha_k\}_{k=1}^{K}$ and K is the number of groups the weights are split into. The approximation to C.1 in this framework can then be written

$$q\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)=q\left(\mathbf{w}\left|D,\theta\right.\right)q\left(\tau\left|D,\theta\right.\right)q\left(\boldsymbol{\alpha}\left|D,\theta\right.\right)$$

where $D=\left\{y_i,\mathbf{x}_i^N\right\}_{i=1}^{N}$. The $\mathcal{KL}$-distance between $p\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)$ and $q\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)$ are

$$\begin{aligned}&\mathcal{KL}\left(p\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right),q\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)\right)=\\&\quad-\int q\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)\log\frac{p\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)}{q\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)}d\mathbf{w},\tau,\boldsymbol{\alpha}\end{aligned}$$

As seen this splits into

$$\begin{aligned}\mathcal{KL}\left(p,q\right)=-\int & q_{\mathbf{w}}q_{\tau}q_{\boldsymbol{\alpha}}(k+\theta\log p_y+\log p_{\mathbf{w}}+\\ &\log p_{\tau}+\log p_{\boldsymbol{\alpha}}-\log q_{\mathbf{w}}-\log q_{\tau}-\log q_{\boldsymbol{\alpha}})d\mathbf{w},\tau,\boldsymbol{\alpha}\end{aligned} \tag{C.2}$$

where $p\equiv p\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)$, $q\equiv q\left(\mathbf{w},\tau,\boldsymbol{\alpha}\left|D,\theta\right.\right)$, $p_y\equiv p\left(\left\{y_i\right\}_{i=1}^{N}\middle|\left\{\mathbf{x}_i\right\}_{i=1}^{N},\tau,\mathbf{w}\right)$, $p_{\mathbf{w}}\equiv p\left(\mathbf{w}\left|\boldsymbol{\alpha}\right.\right)$, $p_{\tau}\equiv p\left(\tau\right)$, $p_{\boldsymbol{\alpha}}\equiv p\left(\boldsymbol{\alpha}\right)$, $q_{\mathbf{w}}\equiv q\left(\mathbf{w}\left|D,\theta\right.\right)$, $q_{\tau}\equiv q\left(\tau\left|D,\theta\right.\right)$, $q_{\boldsymbol{\alpha}}\equiv q\left(\boldsymbol{\alpha}\left|D,\theta\right.\right)$ and k is the negative logarithm of the normalizing constant to C.1.

Derivation of $q_{\mathbf{w}}$

If we carry out the integral over $\boldsymbol{\alpha}$

$$\mathcal{KL}(p,q) = -\int q_{\mathbf{w}} q_{\tau} (k_1 + \theta \log p_y + \int q_{\boldsymbol{\alpha}} \log p_{\mathbf{w}} d\boldsymbol{\alpha} \\ + \log p_{\tau} - \log q_{\mathbf{w}} - \log q_{\tau}) d\mathbf{w}, \tau \quad \text{(C.3)}$$

and over τ

$$\mathcal{KL}(p,q) = -\int q_{\mathbf{w}} \left(k_2 + \theta \int q_{\tau} \log p_y d\tau + \int q_{\boldsymbol{\alpha}} \log p_{\mathbf{w}} d\boldsymbol{\alpha} - \log q_{\mathbf{w}} \right) d\mathbf{w}$$

which equals

$$\mathcal{KL}(p,q) = -\int q_{\mathbf{w}} \left(\log \frac{\exp\left(k_2 + \theta \int q_{\tau} \log p_y d\tau + \int q_{\boldsymbol{\alpha}} \log p_{\mathbf{w}} d\boldsymbol{\alpha}\right)}{q_{\mathbf{w}}} \right) d\mathbf{w}$$

this is minimized by

$$q_{\mathbf{w}} \propto \exp\left(\theta \int q_{\tau} \log p_y d\tau + \int q_{\boldsymbol{\alpha}} \log p_{\mathbf{w}} d\boldsymbol{\alpha} \right)$$

The integrals can be solved

$$\begin{aligned} \int q_{\tau} \log p_y d\tau &= \int q_{\tau} \left(\frac{N}{2} \log \frac{\tau}{2\pi} - \frac{\tau}{2} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 \right) d\tau \\ &= k_{\tau} - \frac{\overline{\tau}}{2} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 \end{aligned}$$

where $\overline{\tau} = \int q_{\tau} \tau d\tau$ i.e. the mean with respect to the approximation q_{τ}. The other integral yields

$$\begin{aligned} \int q_{\boldsymbol{\alpha}} \log p_{\mathbf{w}} d\boldsymbol{\alpha} &= \int q_{\boldsymbol{\alpha}} \sum_{k=1}^{K} \left(\frac{N_k}{2} \log \frac{\alpha_k}{2\pi} - \frac{\alpha_k}{2} \sum_{i=1}^{N_k} \left(w_i^{(k)} \right)^2 \right) d\boldsymbol{\alpha} \\ &= k_{\boldsymbol{\alpha}} - \sum_{k=1}^{K} \frac{\overline{\alpha}_k}{2} \sum_{i=1}^{N_k} \left(w_i^{(k)} \right)^2 \end{aligned}$$

where the prior $p_{\mathbf{w}}$ is independent Gaussians on each weight. The weights are split into K groups, N_k is the number of weights in the k^{th} group and

$w_i^{(k)}$ is the i^{th} weight in that group[1]. $\overline{\alpha}_k = \int q_{\boldsymbol{\alpha}} \alpha_k d\boldsymbol{\alpha}$. So now we can write

$$q_{\mathbf{w}} \propto \exp -\frac{1}{2}\left(\theta\overline{\tau}\sum_{i=1}^{N}\left(y_i - f\left(\mathbf{x}_i, \mathbf{w}\right)\right)^2 + \sum_{k=1}^{K}\sum_{i=1}^{N_k}\overline{\alpha}_k\left(w_i^{(k)}\right)^2\right) \tag{C.4}$$

the exponent is non linear in the weights. For that reason $q_{\mathbf{w}}$ is approximated by a Gaussian, the solution has to be iterated, so in the m^{th} iteration the approximation becomes

$$q_{\mathbf{w}}^{(m)} \propto \exp -\frac{1}{2}\left(\mathbf{w}-\overline{\mathbf{w}}^{(m)}\right)^T \boldsymbol{\Sigma}_w^{(m)^{-1}}\left(\mathbf{w}-\overline{\mathbf{w}}^{(m)}\right) \tag{C.5}$$

The exponent $E = -\log q_{\mathbf{w}}$ in C.4 is approximated by a second order Taylor expansion around $\overline{\mathbf{w}}^{(m-1)}$

$$E(\mathbf{w}) \simeq E\left(\overline{\mathbf{w}}^{(m-1)}\right) + \left.\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m-1)}}\left(\mathbf{w}-\overline{\mathbf{w}}^{(m-1)}\right) + \frac{1}{2}\left(\mathbf{w}-\overline{\mathbf{w}}^{(m-1)}\right)^T \left.\frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}\partial \mathbf{w}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m-1)}}\left(\mathbf{w}-\overline{\mathbf{w}}^{(m-1)}\right)$$

The next things to do is to derive $\overline{\mathbf{w}}^{(k)}$ and $\boldsymbol{\Sigma}_{w^{(k)}}^{-1}$ to do this we only need those parts that are multiplied by $\mathbf{w}$ and find those parts that correspond to those in the exponent of C.5

$$\begin{aligned}&\frac{1}{2}\mathbf{w}^T\boldsymbol{\Sigma}_w^{(m)^{-1}}\mathbf{w} - \mathbf{w}^T\boldsymbol{\Sigma}_w^{(m)^{-1}}\overline{\mathbf{w}}^{(m)} = \\ &\frac{1}{2}\mathbf{w}^T \left.\frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}\partial \mathbf{w}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m-1)}}\mathbf{w} - \mathbf{w}^T\left.\left(\frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}\partial \mathbf{w}^T}^{(m-1)}\overline{\mathbf{w}}^{(k-1)} - \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}\right)\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m-1)}}\end{aligned} \tag{C.6}$$

so by recognition

$$\boldsymbol{\Sigma}_w^{(m)^{-1}} = \left.\frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}\partial \mathbf{w}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m-1)}} \tag{C.7}$$

and

$$\overline{\mathbf{w}}^{(m)} = \overline{\mathbf{w}}^{(m-1)} - \boldsymbol{\Sigma}_w^{(m)} \left.\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m-1)}} \tag{C.8}$$

This ends the approximation of $q_{\mathbf{w}}$.

Derivation of $q_{\boldsymbol{\alpha}}$

The next step is to derive $q_{\boldsymbol{\alpha}}$. As a beginning τ and $\mathbf{w}$ is integrated out of C.3

$$\begin{aligned}\mathcal{KL}(p,q) &= -\int q_{\boldsymbol{\alpha}}\left(k_3 + \int q_{\mathbf{w}}\log p_{\mathbf{w}} d\mathbf{w} + \log p_{\boldsymbol{\alpha}} - \log q_{\boldsymbol{\alpha}}\right)d\boldsymbol{\alpha} \\ &= -\int q_{\boldsymbol{\alpha}}\log\frac{\exp(k_3 + \int q_{\mathbf{w}}\log p_{\mathbf{w}} + \log p_{\boldsymbol{\alpha}})}{q_{\boldsymbol{\alpha}}}d\boldsymbol{\alpha}\end{aligned}$$

[1] In the widely used setup with a group for each layers weight and a group for each layers biases, $K = 4$ if the setup consist of a single layer.

i.e.

$$q_{\boldsymbol{\alpha}} \propto p_{\boldsymbol{\alpha}} \exp \int q_{\mathbf{w}} \log p_{\mathbf{w}} d\mathbf{w}$$

the integral is easily solved

$$\begin{aligned}
\int q_{\mathbf{w}} \log p_{\mathbf{w}} d\mathbf{w} &= \int q_{\mathbf{w}} \sum_{k=1}^{K} \left(\frac{N_k}{2} \log \frac{\alpha_k}{2\pi} - \frac{\alpha_k}{2} \sum_{i=1}^{N_k} \left(w_i^{(k)} \right)^2 \right) d\mathbf{w} \\
&= \sum_{k=1}^{K} \left(\frac{N_k}{2} \log \frac{\alpha_k}{2\pi} - \frac{\alpha_k}{2} \sum_{i=1}^{N_k} \left(\Sigma_{ii}^{(k)} + \overline{w}_i^{(k)^2} \right) \right)
\end{aligned}$$

where (k) indicates the group, and i the number in that group. The result is then

$$\begin{aligned}
q_{\boldsymbol{\alpha}} &\propto p_{\boldsymbol{\alpha}} \exp \sum_{k=1}^{K} \left(\frac{N_k}{2} \log \frac{\alpha_k}{2\pi} - \frac{\alpha_k}{2} \sum_{i=1}^{N_k} \left(\Sigma_{ii}^{(k)} + \overline{w}_i^{(k)^2} \right) \right) \\
&\propto p_{\boldsymbol{\alpha}} \left[\prod_{k=1}^{K} \alpha_k^{\frac{N_k}{2}} \exp -\frac{1}{2} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \alpha_k \left(\Sigma_{ii}^{(k)} + \overline{w}_i^{(k)^2} \right) \right]
\end{aligned}$$

this is simply K independent gamma distributions $q_{\alpha_k} = \frac{1}{\Gamma(a_k) b_k^{a_k}} \alpha_k^{a_k - 1} \exp -\alpha_k / b_k$ so

$$q_{\boldsymbol{\alpha}} = \prod_{k=1}^{K} q_{\alpha_k}$$

where

$$\begin{aligned}
a_k &= a_{p,k} + \frac{N_k}{2} \\
b_k^{-1} &= \left[b_{p,k}^{-1} + \frac{1}{2} \sum_{i=1}^{N_k} \left(\Sigma_{ii}^{(k)} + \overline{w}_i^{(k)^2} \right) \right]^{-1}
\end{aligned}$$

where p indicates the prior, so

$$\overline{\alpha}_k = a_k b_k \tag{C.9}$$

Derivation of q_τ

As in the two preceding derivations, we start by integrating out the variables other than τ from

$$\mathcal{KL}(p, q) = -\int q_\tau \left(k_4 + \theta \int q_{\mathbf{w}} \log p_y d\mathbf{w} + \log p_\tau - \log q_\tau \right) d\tau \tag{C.10}$$

so

$$q_\tau \propto p_\tau \exp \theta \int q_{\mathbf{w}} \log p_y d\mathbf{w}$$

the integral

$$\theta \int q_{\mathbf{w}} \log p_y d\mathbf{w} = \frac{N\theta}{2} \log \frac{\tau}{2\pi} - \theta \int q_{\mathbf{w}} \frac{\tau}{2} \sum_{i=1}^{N} \left(y_i - f\left(\mathbf{x}_i, \mathbf{w}\right)\right)^2 d\mathbf{w}$$

is solved by Taylor expansion

$$\begin{aligned}
\delta_i^2\left(\mathbf{w}\right) &= \left(y_i - f\left(\mathbf{x}_i, \mathbf{w}\right)\right)^2 \\
&\simeq \delta_i^2\left(\overline{\mathbf{w}}^{(m)}\right) + \left.\frac{\partial \delta_i^2\left(\mathbf{w}\right)}{\partial \mathbf{w}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m)}} \left(\mathbf{w} - \overline{\mathbf{w}}^{(m)}\right) + \\
&\quad \frac{1}{2}\left(\mathbf{w} - \overline{\mathbf{w}}^{(m)}\right)^T \left.\frac{\partial^2 \delta_i^2\left(\mathbf{w}\right)}{\partial \mathbf{w} \partial \mathbf{w}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m)}} \left(\mathbf{w} - \overline{\mathbf{w}}^{(m)}\right)
\end{aligned}$$

so

$$\theta \int q_{\mathbf{w}} \log p_y d\mathbf{w} = \frac{N\theta}{2} \log \frac{\tau}{2\pi} - \frac{\tau\theta}{2} \sum_{i=1}^{N} \left(\delta_i^2\left(\overline{\mathbf{w}}^{(m)}\right) - \frac{1}{2} \sum_{jj'=1}^{N_w} \left.\frac{\partial^2 \delta_i^2\left(\mathbf{w}\right)}{\partial w_j \partial w_{j'}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m)}} \Sigma_{jj'}^{(m)} \right)$$

which yields the distribution at iteration m

$$q_\tau^{(m)} \propto \tau^{a_\tau - 1} \exp -\tau / b_\tau^{(m)}$$

i.e. a gamma distribution $\Gamma\left(a_\tau, b_\tau^{(m)}\right)$,where

$$\begin{aligned}
a_\tau &= a_{p,\tau} + \frac{N\theta}{2} \\
b_\tau^{(m)} &= \left[b_{p,\tau}^{-1} + \frac{\theta}{2} \sum_{i=1}^{N} \left(\delta_i^2\left(\overline{\mathbf{w}}^{(m)}\right) + \frac{\tau}{2} \sum_{jj'=1}^{N_w} \left.\frac{\partial^2 \delta_i^2\left(\mathbf{w}\right)}{\partial w_j \partial w_{j'}^T}\right|_{\mathbf{w}=\overline{\mathbf{w}}^{(m)}} \Sigma_{jj'}^{(m)} \right) \right]^{-1}
\end{aligned}$$

so the mean becomes

$$\overline{\tau}^{(m)} = a_\tau b_\tau^{(m)} \tag{C.11}$$

C.2 Algorithm for ensemble learning

The previous sections shows how to calculate $\overline{\tau}$ and $\overline{\boldsymbol{\alpha}}$ if we really know the mean $\overline{\mathbf{w}}$ and $\boldsymbol{\Sigma}_w^{-1}$. But we do not know them, so we hope that it is possible to iterate towards a solution. The algorithm simply becomes

- set $m = 0$
- first draw or select a starting point for $\overline{\mathbf{w}}^{(m)}$

  1. increase k
  2. calculate ${\mathbf{\Sigma}_w^{(m)}}^{-1}$ according to C.7
  3. calculate $\overline{\mathbf{w}}^{(m)}$ according to C.8
  4. calculate $\overline{\alpha}_k^{(m)}$ by using the recent calculated ${\mathbf{\Sigma}_w^{(m)}}^{-1}$ and $\overline{\mathbf{w}}^{(m)}$ and equation C.9
  5. now calculate $\overline{\tau}^{(m)}$ by C.11

- repeat from a until convergence

When convergence is reached a Gaussian approximation to the weights are derived.

Bibliography

[1] A. Gelman and X. L. Meng, *Simulating normalizing constants: From importance sampling to bridge sampling to path sampling*. To appear in Statistical Science, November 1997.

[2] C. H. Bennett, *Efficient estimation of free energy from monte carlo data*, Journal of Computational Physics, 22 (1976), pp. 245–268.

[3] C. M. Bishop, *Neural Network for Pattern Recognition*, Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, 1 ed., 1995.

[4] K. Conradsen, *En Introduktion til Statistik*, vol. 1A-1B of Statistik, IMM, IMM 2800 Lyngby, Denmark, 7 ed., 1995.

[5] D. D. Barber and C. M. Bishop, *Bayesian model comparison by monte carlo chaining*, Advances in Neural Information Processing Systems, 9 (1997).

[6] A. E. Gelfand and D. K. Dey, *Bayesian model choice: asymptotics and exact calculations*, Journal of the Royal Statistical Society - Series B Methodological, 56 (1994), pp. 501–514.

[7] L. K. Hansen, *Bayesian averaging is well-temperated.* Accepted for NIPS99, Denver, November 29 - December 4, 1999.

[8] T. Heskes, *Bias/variance decomposition for likelihood based estimators*, Neural Computation, 10 (1998), pp. 1425–1433.

[9] H. Lappalainen, *Ensemble learning for independent component analysis*, Proceedings of the ICA99, (1999), pp. 7–12. Aussois, France.

[10] D. J. C. MacKay, *Bayesian interpolation*, Neural Computation, 4 (1992), pp. 415–447.

[11] ——, *Developments in probabilistic modelling with neural networks - ensemble learning*, May 1995. Available on the web at http://wol.ra.phy.cam.ac.uk/mackay/README.html.

[12] ——, *Ensemble learning and evidence maximization*, May 1995. rejected by NIPS*95.

[13] ——, *Introduction to monte carlo methods.* A review paper in the proceedings of an Erice summer school, ed M. Jordan, 1996.

[14] X. L. Meng and W. H. Wong, *Simulating ratios of normalizing constants via a simple identity: a theoretical exploration*, Statistica Sinica, 6 (1996), pp. 831–860.

[15] R. M. Neal, *Probabilistic inference using markov chain monte carlo methods*, Technical Report CRG-TR-93-1, University of Toronto, Department of Computer Science, University of Toronto, September 1993.

[16] ——, *Bayesian Learning for Neural Networks*, vol. 118 of Lecture Notes in Statistics, Springer-Verlag, New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA, 1 ed., 1996.

[17] M. A. Newton and A. E. Raftery, *Approximate bayesian inference and the weighted likelihood bootstrap (with discussion)*, Journal of the Royal Statistical Society - Series B Methodological, 56 (1994), pp. 3–48.

[18] C. E. Rasmussen, *Documentation for the supic neural network package.* Draft intern, June 1998. supic is a Maximum Likelihood based C++ software for Neural Networks.

[19] C. P. Robert, *The Bayesian Choice, A Decision-Theoretic Motivation*, ST Springer Texts in Statistics, Springer-Verlag, New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA, 3 ed., 1994.

[20] T. J. DiCiccio and R. E. Kass and A. Raftery and L. Wasserman, *Computing bayes factors by combining simulation abd asymptotic approximations*, Journal of the American Statistical Association, To appear (1997).