Aalto University
School of Science
Department of Neuroscience and Biomedical Engineering

Tuomas Sivula

Distributed Bayesian inference using expectation propagation

Master's Thesis
Espoo, 22nd April 2015

Supervisor and advisor: Professor Aki Vehtari

Aalto University
School of Science

ABSTRACT OF THE
MASTER'S THESIS

Author: Tuomas Sivula

Title: Distributed Bayesian inference using expectation propagation

Date: 22nd April 2015 **Pages:** 75

Department of Neuroscience and Biomedical Engineering

Professorship: Computational Science **Code:** Becs-114

Supervisor and advisor: Professor Aki Vehtari

This thesis studies, how expectation propagation (EP) can be used in distributed Bayesian inference. The method is discussed in general level and an implementation using normal approximation is presented. In addition, the method is considered in the context of hierarchical probability models.

The EP method is analysed based on source literature. In addition, the method is tested in several simulated hierarchical experiments.

Various methods for distributed Bayesian inference has been developed recently. However, they all perform the inference for each part of the data set independently. In EP, the parts are processed in iterative fashion and the message passing feature distributes the essential information between the parts.

Previously in the EP methods, the data set has usually been factorised pointwise. By distributing the data set into bigger groups, the method can be utilised more versatilely. In hierarchical models, it is beneficial to partition the data set among the hierarchical groups.

The experiments show that the method can produce good results. It can be seen from the results that increasing the number of distributed groups increase the approximation error. In addition, it can be seen that other sources of error affect the results and can prevent the algorithm from converging.

Further research is required for the analysis of the approximation error. In addition, the method should be compared to other distributed Bayesian inference methods.

Keywords: Bayes, EP, data partitioning, parallel, MCMC, hierarchical, Stan

Language: English

Aalto-yliopisto
Perustieteiden korkeakoulu

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Tuomas Sivula		
Työn nimi:	Hajautettu Bayesilainen mallintaminen välittämällä odotusarvoa		
Päiväys:	22. huhtikuuta 2015	**Sivumäärä:**	75
Neurotieteen ja lääketieteellisen tekniikan laitos			
Pääaine:	Laskennallinen tiede	**Koodi:**	Becs-114
Valvoja ja ohjaaja:	Professori Aki Vehtari		

Tämä diplomityö käsittelee odotusarvon välittämismenetelmän (expectation propagation, EP) soveltamista hajautettuun Bayesilaiseen mallintamiseen. Työssä esitetään menetelmän yleinen toimintaperiaate ja yksityiskohtaisempi toteutus normaalijakauma-approksimaatiolle. Lisäksi työssä tutkitaan menetelmän soveltuvuutta hierarkkisiin todennäköisyysmalleihin.

Tutkittavaan menetelmään perehdytään työssä lähdeaineiston pohjalta analysoiden. Lisäksi sovellettua menetelmää testataan simuloiduissa hierarkkisissa testitilanteissa.

Erilaisia hajautettuja menetelmiä, joilla sovitetaan suuria datajoukkoja todennäköisyysmalleihin, on kehitetty viimeaikoina. Niille kaikille on kuitenkin yhteistä se, että kukin datajoukon osa käsitellään erikseen muista riippumattomasti. EP-menetelmässä osia käsitellään iteroiden ja viestin välitys -ominaisuus jakaa keskeistä informaatiota osien välillä.

Aikaisemmin EP-menetelmässä käsiteltävä datajoukko on jaettu tyypillisesti pisteittäin. Jakamalla datajoukko suurempiin osiin, saadaan menetelmää hyödynnettyä monipuolisemmin. Hierarkkisissa malleissa on hyödyllistä jakaa datajoukko ryhmittäin.

Hajautettu EP-menetelmä osoittautui simuloiduissa testitilanteissa toimivaksi. Tuloksista on havaittavissa, että datajoukon osituksen osien lukumäärän kasvattaminen kasvattaa approksimaation virhettä. Lisäksi on ilmeistä, että menetelmässä on myös muita virhelähteitä, joiden vaikutuksesta algoritmin konvergoituminen voi estyä. Menetelmän virhelähteiden arviointi ja vertailu muihin vastaaviin menetelmiin vaativat lisätutkimuksia.

Asiasanat:	Bayes, EP, datan ositus, rinnakkainen, MCMC, hierarkkinen, Stan
Kieli:	Englanti

Acknowledgements

I wish to thank my supervisor and advisor Aki Vehtari for the oppoturnity to work with this topic and for the guidance in the research. I also thank my coworkers for the supportive and friendly work environment.

Espoo, 22nd April 2015

Tuomas Sivula

Contents

Abbreviations and notation

CV	cross-validation
EP	expectation propagation
KL divergence	Kullback-Leibler divergence
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MSE	mean squared error
OLSE	optimal linear shrinkage estimator

$\mathcal{S}_x$	sample space of random variable x
$x \sim \mathrm{X}$	x has the probability distribution X
$x \vert y \sim \mathrm{X}$	x given y has the probability distribution X
$\Pr(\cdot)$	probability of an event
$\mathrm{p}(\cdot)$	probability density or mass function
$\mathrm{p}(x \vert y)$	conditional probability density or mass function, if seen as a function of x for given y, or likelihood function, if seen as a function of y for given x
$\mathrm{E}(\cdot)$	expectation
$\mathrm{E}(\cdot \vert \cdot)$	conditional expectation
$\mathrm{Var}(\cdot)$	variance, if operated on a one dimensional random variable, or covariance matrix, if operated on a multidimensional random variable
$\mathrm{Cov}(\cdot, \cdot)$	covariance (or cross-covariance)
$\mathrm{Cor}(\cdot, \cdot)$	correlation coefficient (or cross-correlation)

Bernoulli(p)	Bernoulli distribution with success probability p
Beta(α, β)	beta distribution with respective parameters
half-Cauchy(μ, σ)	Cauchy distribution with location μ and scale σ restricted to be greater than μ
Laplace(μ, σ)	Laplace (also known as double exponential) distribution with location μ and scale σ

$\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ (or alternatively with natural parameters $\boldsymbol{\eta}$ and $\boldsymbol{\Omega}$)
$\mathrm{N}(\mu, \sigma)$	normal distribution with mean μ and standard deviation Σ
log-$\mathrm{N}(\mu, \sigma)$	log-normal distribution, that is $x \sim \text{log-N}(\mu, \sigma) \Rightarrow \log(x) \sim \mathrm{N}(\mu, \sigma)$
logit-$\mathrm{N}(\mu, \sigma)$	logit-normal distribution, that is $x \sim \text{logit-N}(\mu, \sigma) \Rightarrow \mathrm{logit}(x) \sim \mathrm{N}(\mu, \sigma)$
$\mathrm{X}(\cdot\,\vert\theta)$	probability density function of a distribution X with given parameters θ
$\mathbf{I}$ or $\mathbf{I}_n$	the identity matrix (of size n)
$\mathbb{1}$ or $\mathbb{1}_n$	column vector of ones (of length n)
$\mathrm{erf}(\cdot)$	Gauss error function, that is $\mathrm{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^{\pi} \mathrm{e}^{-t^2}\, \mathrm{d}t$
$\mathrm{logit}(\cdot)$	logit function, that is $\mathrm{logit}(p) = \log\left(\frac{p}{1-p}\right)$, where $p \in [0, 1]$
$\mathrm{nul}(\cdot)$	matrix nullity
$\mathrm{rank}(\cdot)$	matrix rank
$\mathrm{tr}(\cdot)$	matrix trace
$\lVert\boldsymbol{A}\rVert_{\mathrm{F}}$	Frobenius norm of matrix $\boldsymbol{A}$, that is $\lVert\boldsymbol{A}\rVert_{\mathrm{F}} = \sqrt{\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})}$
$\lVert\boldsymbol{A}\rVert_{\mathrm{tr}}$	trace norm of matrix $\boldsymbol{A}$, that is $\lVert\boldsymbol{A}\rVert_{\mathrm{tr}} = \mathrm{tr}\left(\sqrt{\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}}\right)$, where $\sqrt{\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}} = \boldsymbol{B}$, such that $\boldsymbol{B}\boldsymbol{B} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$

Chapter 1

Introduction

When dealing with large data sets, the computational complexity of data analysis algorithms often grow into intolerable magnitudes. Partitioning the data and processing each part in a parallel fashion is a general scalable approach for coping with this deficiency. However, such a partitioned analysis usually introduces approximation error in the analysis results. Parallelisation in general is a popular research topic at the moment.

Distributed inference in a context of Bayesian probabilistic modeling is not trivial. Various methods for fitting probabilistic models to large data sets have been proposed. The general workflow in all of them is the following: split the data, perform inference separately for each part and combine the results. In such a method, the inference for each separate unit is performed independent of the others while in reality they should also consider information from the others. This thesis reviews and experiments a method that performs the inference for each part iteratively and shares the information from the other in between.

Expectation propagation (EP) is a parallelisable distributional method for performing approximate Bayesian inference. Usually this method is used to distribute each individual data point into separate part commonly referred to as sites. In such a case, the inference on the separate sites often becomes easyer. Gelman et al. (2014a) propose to use EP so that the partitioning is not done pointwise but in larger parts. In this case, the site inference becomes more complex but also more informatic. This approach provides more flexible distributed method for Bayesian inference. Furthermore, in the case of hierarchical models, the method can be simplified by distributing each group to separate sites. In this thesis, the term distributed EP refers to the EP method proposed by Gelman et al. (2014a).

This thesis presents the distributed EP method and discusses various aspects and additional features related to it. Several simulated data problems are used to test the method. Because the experimented problems are relatively small, the scalability of the method into bigger problems requires further analysis. This thesis does not discuss or analyse algorithmic aspects related to the time or

memory complexity of the presented EP algorithm. Furthermore, the performance of the distributed EP method is not compared against other distributed Bayesian inference methods. In addition, a couple of variance reduction methods usable for the algorithm are presented and experimented, but further analysis would be required to make general conclusions about their applicability in this problem.

Introduction to the Bayesian inference problem and into the distributed approach is presented in chapter 2. Chapter 3 presents the EP method in a general level and chapter 4 applies it into distributed context. The advantages of applying the distributed EP method to hierarchical models is presented in section 4.3. Various additional considerations and enhancements to the method are discussed in chapter 5. In chapter 6, the setup for the experiments and the results for them are presented. Finally, in chapter 7 the distributed EP method and the experiment results are analysed together more deeply.

Chapter 2

Bayesian computation in large scale

This chapter presents the basic intuition behind Bayesian inference and computational methods related to it. The inference problem is introduced in section 2.1 and the ultimate reason for using approximative methods, analytically unsolvable integrals, is described in section 2.2. Furthermore, section 2.3 discusses the computational complexity when working with large data sets and presents some methods for dealing with this increased burden.

2.1 Bayesian inference

Statistical inference is the process of learning properties of unobserved quantities or random variables based on a related sample data. In order to make these probabilistic deductions, the observations and the quantities of interest are assumed to follow some underlying statistical model. In Bayesian approach, Bayes' theorem is incorporated into the process by conditioning on the observed data. The essential feature on these types of methods is that they directly quantify the uncertainty of the inference probabilistically.

Bayesian data analysis process in general does not consist only of the inference part. The whole process includes also the initial definition step, where the model assumptions are assigned, and the final validation step, where the goodness of the fit is evaluated. The focus of this thesis is, however, not in these two steps, but in the step between: the evaluation of the posterior distribution.

Posterior distribution

Let $\theta \in \mathcal{S}_\theta$ and $y \in \mathcal{S}_y$ be jointly distributed and possibly multidimensional random variables, where θ denote an unknown unobserved quantity or parameter, y denote the observed data and $\mathcal{S}$ denote the respective sample space. The goal is to perform inference for θ conditional on y by finding the posterior distribution

according to the Bayes' theorem

$$\mathrm{p}(\theta|y) = \frac{\mathrm{p}(y|\theta)\,\mathrm{p}(\theta)}{\mathrm{p}(y)}, \tag{2.1}$$

where $\mathrm{p}(y|\theta)$ is the likelihood function, $\mathrm{p}(\theta)$ is the prior and $\mathrm{p}(y)$ is the normalising term sometimes referred as evidence. The likelihood is regarded as a function of θ for given y. The normalising term is given by

$$\mathrm{p}(y) = \begin{cases} \int_{\mathbb{S}_\theta} \mathrm{p}(y|\theta)\,\mathrm{p}(\theta)\,\mathrm{d}\theta & \text{if } \theta \text{ is continuous,} \\ \sum_{\theta\in\mathbb{S}_\theta} \mathrm{p}(y|\theta)\,\mathrm{p}(\theta) & \text{if } \theta \text{ is discrete.} \end{cases} \tag{2.2}$$

From here on, unless otherwise noted, equations are only presented for continuous variables and the analogous forms for discrete cases are omitted. Also, when notating a definite integral over the whole sample space of a random variable, the self-evident domain is left out.

Inference target

In addition to the unknown and unobservable parameters of the model, the variable of interest can also be an unknown but observable future sample $\widetilde{y}$. In this case, the goal is to find the posterior predictive distribution of $\widetilde{y}$ by

$$\mathrm{p}(\widetilde{y}|y) = \int \mathrm{p}(\widetilde{y}|\theta, y)\,\mathrm{p}(\theta|y)\,\mathrm{d}\theta. \tag{2.3}$$

If $\widetilde{y}$ and y can be assumed to be conditionally independent given θ, the equation simplifies further into $\mathrm{p}(\widetilde{y}|y) = \int \mathrm{p}(\widetilde{y}|\theta)\,\mathrm{p}(\theta|y)\,\mathrm{d}\theta$. This type of predictive inference is not discussed further in this thesis.

Moreover, it is common that the posterior distribution itself is not the final objective, but further inference upon this distribution is required. Often a multidimensional posterior distribution needs to be marginalised over nuisance parameters or summarised by moments. Typically this analysis consist of an evaluation of an integral of the form

$$\mathrm{E}\big(g(\theta)\big|y\big) = \int g(\theta)\,\mathrm{p}(\theta|y)\,\mathrm{d}\theta, \tag{2.4}$$

where $g(\theta) : \mathbb{S}_\theta \to \mathbb{R}$ is some function of interest. For example, defining $g(\theta) = (\theta - c)^k$ yields the k-th posterior moment about a value c. Combining from equations (2.1), (2.2) and (2.4), it can be seen that the full inference requires calculating a ratio of integrals

$$\mathrm{E}\big(g(\theta)\big|y\big) = \frac{\int g(\theta)\,\mathrm{p}(y|\theta)\,\mathrm{p}(\theta)\,\mathrm{d}\theta}{\int \mathrm{p}(y|\theta)\,\mathrm{p}(\theta)\,\mathrm{d}\theta}. \tag{2.5}$$

2.2 Finding the posterior

As seen in the previous section, the posterior distribution is the cornerstone of all subsequent inference summaries in Bayesian data analysis. However, except in some rather constrained cases, the posterior distribution can not be solved analytically in a closed form. Exceptions include cases where the likelihood is of some specific form and the prior distribution is conjugate to the likelihood. Because of this, the inference often has to be carried out approximately, either by numerical integration of distributional approximation.

Numerical methods

Approximating integrals is not trivial and approximating multidimensional integrals, as is often the case in Bayesian inference, is even harder. Numerical methods designed for these problems can be divided into two groups: simulation methods and deterministic methods. Simulation methods, such as direct simulation or Monte Carlo (MC), are based on obtaining posterior samples and estimating the integral stochastically using these samples. Deterministic methods, such as grid integration or most quadrature methods, are based on evaluating the integral at finite set of points and combining the densities.

Markov chain Monte Carlo (MCMC) methods are popular sampling techniques, where the samples are drawn sequentially based on the previous samples. By nature, these methods produce dependent samples, but the dependence can be countered for example by thinning, that is discarding samples periodically, and by running multiple independent chains. Here it is also important to run the chains long enough and discard a portion of samples from the beginning of each chain so that the starting points do not influence the results. When compared to the basic MC methods, which produce independent samples, the MCMC methods can better adapt to high-dimensional distributions. The most general MCMC methods are Metropolis (Metropolis and Ulam 1949; Metropolis et al. 1953) and Metropolis–Hastings (Hastings 1970). The Stan computation environment used in the experiments of this thesis use Hamiltonian MCMC method (Duane et al. 1987; Neal 1994) with no-U-turn sampler (Homan and Gelman 2014).

Distributional methods

Distributional approximations are analytic methods that try to approximate the true posterior with some simpler distribution, from which the desired inferences can be calculated directly or numerically more efficiently. Because the computational complexity of some of these methods is lower than of the numerical methods in general, these can be used to get a crude approximation usable as a starting

point for sampling based methods.

A straightforward way to perform distributional approximation is to fit a (multivariate) normal distribution by centring it to the mode of the posterior and scaling it based on the curvature at that point. A second order Taylor expansion of the target log probability density function around the mode shows that the optimal covariance matrix equals to the inverse of the Hessian of the negative log probability density function at that point. Laplace's method approximates the conditional expectation in equation (2.4) directly based on this normal approximation (Tierney, Kass and Kadane 1989; Tierney and Kadane 1986). Naturally, these methods require that the posterior mode can be found first. Usually this is conducted with some iterative search algorithm, such as a simple conditional maximisation, the Newton's method or the Broyden–Fletcher–Goldfarb–Shanno quasi-Newton method. The first and the second methods are discussed for example by Gelman et al. (2014b, p. 312) and the last one is reviewed for example by Al-Baali, Spedicato and Maggioni (2014). If the posterior distribution is multimodal, mixture approximations can be used to enhance these methods (Gelman et al. 2014b, p. 319).

Laplace's method can be further extended to use split normal approximation in order to better capture possible skewness of the target distribution. The method for fitting this distribution is described by Geweke (1989, pp. 1324-1326) and the distribution itself along with necessary properties are defined by Villani and Larsson (2006). The main idea is to scale each principal component direction in the approximation by exploring the rate of decline in the corresponding direction from the mode in the target density. Instead of normal distribution, other distributions can also be used in the split context. Geweke (1989, pp. 1324-1326) introduces also an alternative robust method using student's t-distribution.

Other more advanced distributional approximation methods include, for example, variational Bayes (Beal 2003) and expectation propagation (EP) (Minka 2001b). Both are iterative methods that approximate the target distribution with some factorised and constrained distribution. In the former method, the problem is usually partitioned into the components of the parameter vector θ, whereas in the latter, the partitioning is usually done for the data y. As EP, and its message passing approach in particular, are fundamental features in the distributed inference method described in this thesis, those are discussed more deeply in section 3.

More robust and efficient methods can be developed by combining different simulation, deterministic and distributional methods. The topic of this thesis, the distributed EP method, is one such a technique. It combines aspects from EP with sampling or alternatively with some other distributional approximation method.

2.3 Data partitioning

When considering the described Bayesian inference problem, it is obvious that the computational burden increases, depending on the method, at least linearly along with the number of data samples. Thus, as in so many other data analysis problems, it would be highly beneficial to be able to perform the inference in a distributed fashion, when dealing with large data sets.

It can be seen from equations (2.1) and (2.2), that the data affects the inference only through the likelihood. If the observed data points y are assumed independent conditional on the model parameters θ, the likelihood can be conveniently factored as

$$\mathrm{p}(y|\theta) = \prod_{i=1}^{K} \mathrm{p}(y_i|\theta), \tag{2.6}$$

where y has been partitioned into K subsets $y_1, y_2, \ldots, y_K$. An intuitive approach to distributing the inference would be to analyse each of these likelihood contributions separately and combine the resulting individual posteriors. However, even though the likelihood factorises naively, the posterior distribution does not.

Performing such a distributed inference on a factored likelihood requires, that the prior distribution is also propagated into each inference unit somehow. One option is to associate each likelihood factor with $\mathrm{p}(\theta)^{1/K}$, so that their product is the full unnormalised posterior distribution. This can, however cause problems, if an informative prior is necessary for good estimation of θ; with large K, the effect of $\mathrm{p}(\theta)^{1/K}$ becomes increasingly small. Another option is to use the full prior for each inference and divide by $\mathrm{p}(\theta)^{K-1}$ at the combination stage. Nevertheless, this approach may also cause numerical instability because of the final normalising division.

Different techniques has been proposed recently for performing a distributed Bayesian inference (Ahn, Korattikara and Welling 2012; Gershman, Hoffman and Blei 2012; Hoffman et al. 2013; Korattikara, Chen and Welling 2014; Neiswanger, Wang and Xing 2014; Scott et al. 2013; Wang and Blei 2013; Wang and Dunson 2014). While all of these methods incorporate different assumptions and approximations, they all share the same previously described workflow: divide the data set into pieces, perform the inference separately on all the pieces and combine the results. Here it is noteworthy, that when the inference for each piece is performed independently from the others, no data related information is shared among the inference units before the final combination stage.

In an ideal distributed Bayesian inference, each unit should get feedback from other units and adjust their decision making based on the others. A simple way to introduce this message passing behaviour into an existing distributed inference method, is to iterate it while using the posterior of the previous iteration as the prior for the new iteration. This way the information from the other units

is included in the inference after the first initial iteration. Somewhat similar approach is used in the EP method, which is discussed in more detail in chapter 4.

Chapter 3

Expectation propagation

The distributed Bayesian inference method described in this thesis incorporates the EP algorithm as a way to share information between individual inference units. As discussed in section 2.3, this message passing approach is a key feature in improving a naive one pass distributed inference method. The EP algorithm was briefly mentioned in section 2.2 together with some other distributional approximation methods but this chapter presents it in more detail.

Section 3.1 presents the general EP framework and its iterative algorithm. Approximating the tilted distribution is a key step in the algorithm. This approximation and different approaches into it are discussed in section 3.2. Furthermore, in section 3.3, the problem of convergence and stability is addressed.

3.1 General framework

Expectation propagation (EP) is an iterative deterministic method developed initially by Opper and Winther (2000) and later more generally by Minka (2001a,b). It approximates a target probability distribution with some exponential family distribution. In a more general setting, other distributions can also be used, although this complicates the calculations and denies some theoretical results related to the convergence. In Bayesian inference, EP is often used to approximate an intractable posterior distribution $\mathrm{p}(\theta \,|\, y)$.

Let $f(\theta)$ denote the target distribution with some convenient factorisation into K sites

$$f(\theta) \propto \prod_{i=0}^{K} f_i(\theta)\,. \tag{3.1}$$

Here it is not necessary for the the factors $f_i(\theta)$ to be probability distributions. They can be for example unnormalised distributions or likelihood functions. Usually this method is applied in Bayesian context so that each factor corresponds to the likelihood of a single data point. However, other factorisations can be applied, for example, by combining multiple data points together. Let $g(\theta)$ denote the

approximating distribution

$$g(\theta) \propto \prod_{i=0}^{K} g_i(\theta), \tag{3.2}$$

where $g(\theta)$ and each $g_i(\theta)$ follow some distribution of choice from the exponential family. Each site term $g_i(\theta)$ approximates corresponding factor from equation (3.1).

The main idea of the algorithm is to iteratively update each site term $g_i(\theta)$ by approximating the respective true site term in the context of the current global approximation. Let cavity distribution $g_{-i}(\theta)$ denote the product of all the other $K-1$ site approximations except the currently selected one

$$g_{-i}(\theta) \propto \prod_{j \neq i} g_j(\theta)\,. \tag{3.3}$$

The tilted distribution $g_{\setminus i}(\theta)$ is formed by replacing the site approximation with the corresponding factor $f_i(\theta)$ in the global approximation, that is

$$g_{\setminus i}(\theta) \propto f_i(\theta) \prod_{j \neq i} g_j(\theta)\,. \tag{3.4}$$

This distribution includes the correct factor for the current site and approximations for the others. In each iteration, the site approximation $g_i(\theta)$ is updated by fitting the global approximation $g(\theta)$ into the tilted distribution.

In the beginning of the algorithm, some initial distributions are chosen for the site approximations and the global approximation is calculated from equation (3.2). Then the following EP algorithm is iterated until the global approximation converges. One EP iteration consist of updating each site approximation once, that is for $i = 1, 2, \ldots, K$:

1. Compute the cavity distribution

$$g_{-i}(\theta) \propto \frac{g(\theta)}{g_i(\theta)}. \tag{3.5}$$

   Here it should be noted that the cavity distribution can also be evaluated from equation (3.3), but often equation (3.5) provides more numerically efficient way. When working with the natural parameters of the normal distribution, product of distributions can be conducted by summation of the parameters and division by subtraction. Thus, using equation (3.3), the calculation of the cavity distribution is carried out by $K-1$ summations for each site, whereas using equation (3.5), only one sutraction is needed for each site. The latter approach has worse precision however, because one of the terms is summend and later subtracted away. Although probably not beneficial, other more complex and memory consuming approaches can be taken by combining summations of common terms in different sites. These implementational aspects are discussed more deeply in chapter 4.

2. Form the tilted distribution

$$g_{\setminus i}(\theta) \propto f_i(\theta)\, g_{-i}(\theta)\,. \tag{3.6}$$

3. Approximate the tilted distribution with a distribution $\widetilde{g}(\theta)$ from the chosen family.

4. Update the site approximation

$$g_i(\theta) \propto \frac{\widetilde{g}(\theta)}{g_{-i}(\theta)}. \tag{3.7}$$

5. If serial, update the global approximation

$$g(\theta) = \widetilde{g}(\theta) \propto g_i(\theta)\, g_{-i}(\theta)\,. \tag{3.8}$$

If the algorithm is run in serial, the global approximation $g(\theta)$ is updated after each site approximation update in the step 5. Otherwise, if the algorithm is run in parallel fashion, the global approximation is updated only after all of the site approximations has been updated, that is after all the steps 1–4 are run for each $i = 1, 2, \ldots, K$. The former method is called sequential EP and the latter parallel EP.

As mentioned before, in the case of Bayesian computing, the site terms in the target distribution have usually been partitioned among the data points, that is each site term $f_i(\theta)$ corresponds to the likelihood of one data point. In addition, one of the site terms are assigned into the prior distribution. Naturally if the prior distribution is already of the required form, the corresponding site approximation can be set to the precise distribution from the beginning and it does not need to be updated during the iteration procedure. Instead of the usual way of partitioning all the data points into separate sites, the distributed method described in this thesis assigns multiple points into one site.

3.2 Approximating the tilted distribution

The approximation of the tilted distribution in the step 3 of the algorithm can be done in many ways. The situation is quite similar to the problem that the EP algorithm is trying to solve in the first place. However, as only one of the factors in the tilted distribution is difficult to handle, the problem is often more easily approachable.

In the standard EP algorithm, the approximation of the tilted distribution is done by matching the respective moments of the approximative distribution to the ones obtained from the tilted distribution. This requires an integration over a possibly high-dimensional space $\mathcal{S}_\theta$. In many situations, this can be done

analytically in closed form or a dimension reduction method can be utilised (Minka 2001a). If this is not possible or reasonable, various alternative methods exists. Some of these methods and their general approaches are presented in the following sections 3.2.1 and 3.2.2.

3.2.1 Distributional methods

One approach for approximating the tilted distribution is to use Laplace's approximation to match the mode and curvature instead of the moments. Even if the corresponding factor $f_i(\theta)$ does not have an easily identified mode, the presence of the cavity distribution ensures that at least one such exists in the tilted distribution. The mean of the cavity distribution is a good starting point for the mode searching algorithm. Smola, Vishwanathan and Eskin (2004) shows that a solution obtained by this Laplace propagation method is also one possible solution of the Laplace's approximation in the joint model.

Furthermore, here it could be beneficial to use the split normal or split-t approximation discussed in section 2.2 in order to improve the approximation on skewed distributions. Here, as with other distributional tilted distribution approximations, it is not necessary that the family of the approximative distribution $g(\theta)$ is the same as the one used to approximate the tilted distribution. The moments of $g(\theta)$ can be matched with the ones obtained by this approximation. Further improvements can be made for example by using importance sampling (Geweke 1989).

Another interesting possibility is to use EP again to estimate the moments of the tilted distribution (Riihimäki, Jylänki and Vehtari 2013). This nested EP method is applicaple, if the target site terms have some spesific form, for example they are further factorisable with some transformation on the parameter θ. In these situations the inner EP algorithm converges quickly and often only one EP iteration is needed.

3.2.2 Simulation methods

More universally applicable method for approximating the tilted distribution is to use sampling for calculating the moments. The accuracy of the moment estimates is particularly important with EP, as inaccurate estimates may cause instability (Jylänki, Vanhatalo and Vehtari 2011). However, low precision in these estimates may still produce good results in some cases, as can be seen in the results of the experiments in section 6.5. Clearly the model and the method affect this behaviour.

Approximating the moments from samples efficiently is not trivial. The number of required samples for accurate covariance matrix estimation increase heavily

along with the number of dimensions, while the computational burden per sample also increases. As a result, the time complexity may easily explode into intolerable magnitudes, or inaccurate results may cause the EP algorithm to fail. Nevertheless, many methods exists for improving these estimates. Section 5.3 presents some of these methods in more detail.

3.3 Convergence and stability

Running the EP algorithm by moment matching minimises the Kullback-Leibler divergence (KL divergence) (Kullback and Leibler 1951) from the tilted distribution to the corresponding approximating distribution iteratively. However, this does not ensure that the KL divergence from the resulting global approximation to the true target distribution is minimised (Bishop 2006, p. 510). The following results are shown by Minka (2001b). If the approximating distribution is from the exponential family, and if the algorithm converges, the solution will be a stationary point in a particular energy function. Here it is also noteworthy that, if the EP algorithm is run in serial, the behaviour of the algorithm depends on the order in which the sites are processed.

The convergence problems of EP is a considerable feature that has to be payed attention to. One remarkable problem is that when updating the global approximation either in the step 5 of the EP algorithm or after each EP iteration, the resulting distribution does not exist and the algorithm fails. Generally this problem appears in a situations, where the variance or the covariance matrix of the global approximation is not positive or positive definite respectively. A simple solution to this problem is to force the variance positive (Minka 2001a, p. 22) or the matrix to positive definite (Betancourt 2013). Alternatively one can perform only partial site updates by introducing damping. An implementation for damping is presented in section 4.2 and its effects are discussed in section 5.1. Constraining positive definiteness is discussed in more detail in section 5.2.

Another problem that often co-occurs with non-positive variances or covariance matrices, that are not positive definite, is that the site approximations oscillate without ever converging. If the magnitude of the oscillation is small, the problem does not matter, but otherwise, some actions has to be taken in order to gain useful, interpretable results. While damping may help avoiding also this problem, a more direct approach is to smooth the tilted distribution approximations by averaging over few approximations from the previous iterations. Various approaches for performing this weighted moment update are presented in section 5.2

While there is no guarantee that the EP algorithm converges when iterating infinitely, it has been shown to perform well on many practical applications. For example, it performs better or at least as good as both local variational methods

and the Laplace's approximation in logistic regression and classification models (Kuss and Rasmussen 2005). On the other hand, the nature of EP does not always comport with certain type of models; for example when applied to multimodal mixture distributions, EP often yields a solution that is something between all the mixtures, while a solution focused to only one mode could be more useful. Multimodal and long tailed distributions often has stability issues in the tilted distribution approximation step of the EP algorithm.

Many extension methods, that address to the problem of convergence or to the possible computational instabilities, has been developed. Fractional EP is one such method (Minka 2004). As a generalisation to the KL divergence, it corresponds to minimising the α-divergence (Cichocki and Amari 2010), with different choices of α. This method can be used to improve the robustness of the EP algorithm, as the error in the tails of the approximation of the tilted distribution causes less stability issues when $\alpha < 1$.

Chapter 4

Distributed EP

This chapter presents a distributed Bayesian inference method that utilises the message passing feature of the EP method to share information between parallel inference units. Section 4.1 considers EP from distributional point of view in general and presents some approaches related to it. An implementation for distributed EP using normal approximation is described in section 4.2. Furthermore, applying the distributed EP into hierarchical models are discussed in section 4.3. Using hyperparameters complexifies the EP calculations slightly. Dealing with them is briefly discussed in section 4.4.

4.1 Distributed approach to EP

As mentioned before in section 3.1, the factorisation of the likelihood in the EP algorithm in equation (3.1) is typically performed pointwise, that is each site term corresponds to the likelihood of one data point. This factorisation corresponds to ultimate distribution of the data into separate inference units. Instead of this full factorisation, the data set $\boldsymbol{y}$ can also be partitioned into sets consisting of multiple points, where each set is assigned into one EP site. Let N denote the number of data points and K the total number of sites. In general, K can be anything between 1 and N, where the former corresponds to the full inference case without any EP and the latter into the typical pointwise EP inference. Here the prior distribution is not counted into the total number of sites.

Let $\boldsymbol{y} = \left[\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K\right]$ denote the selected partitioning of the data into K subsets. Each site has its own likelihood $\mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta})$. The EP method presented in section 3.1 can be applied to construct an approximation $g(\boldsymbol{\theta})$ for the posterior distribution $\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{y}) \propto \mathrm{p}(\boldsymbol{\theta}) \prod_{k=1}^{K} \mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta})$. The prior for $\boldsymbol{\theta}$ is assigned into one site term, that is kept constant. Here the approximation of the tilted distribution corresponds to a reduced inference problem considering only a part of the data in the likelihood and the corresponding cavity distribution as the prior. The original problem considering the whole data y is distributed into K parallelisable iterated

subproblems.

In a pointwise partitioning, the resulting tilted distributions can often be approximated easily, as only one data point affects them directly. In such situations, the tilted distributions can often be approximated analytically or using one dimensional numerical quadrature. When the number of sites is decreased, the number of assigned data points increase and the inference is likely to become more and more difficult relative to the $K = N$ case. In such a case, the tilted distribution approximation generally does not have analytic solution and other methods for approximating it has to be applied. However, while the site inferences become harder, the amount of information gained in each inference is increased and the total number of required EP iterations is likely to decrease. This trade-off situation implies that balancing between both extremes $K = N$ and $K = 1$ might provide good results.

The computational advantage of using EP in distributed Bayesian inference, when compared to other relative inference methods mentioned in section 2.3, lies in the message passing feature. In distributed Bayesian inference setting, each partition has only a confined view of the global information. If a partition has likelihood contribution in areas that are contradicted by the other $K-1$ partitions, and the inference for this partition is made independent of the other partitions, a lot of computational effort is wasted on this redundant area. Figure 4.1 illustrates this with a simple example. In distributed EP, the information from the other partitions is included as a prior for the inference for the other partitions via the cavity distribution, and the redundancy of the area is taken into account, when performing the inference on that partition. This additional information available after the first iteration may increase the effectiveness of the algorithm.

Depending on the problem, the partitioning of the data may affect the result and the convergence speed of the EP algorithm. Some data sets may have a natural partitioning, which results in increased effectiveness, and breaking this partitioning may cause problems. Even if the number of sites is kept constant but the associated data points are changed, the results may change. Consider a big data set, which is distributed into two sites. Only two points are included into one site and the rest into the other. Clearly it is likely that the partitioning is redundant because the full model considering all the points can be fitted with nearly the same effort as the inference on the bigger site in the EP algorithm in one iteration. The matter of partitioning is clearly a complex problem.

The general EP method can be performed in serial or parallel fashion. Consider a case, where the parallel EP algorithm is run in K parallel computing client units or nodes which are controlled by a single master node. Each site is assigned into one client node, where the relating parallelisable computations, such as the tilted distribution approximation, are carried out. If each site contains multiple data

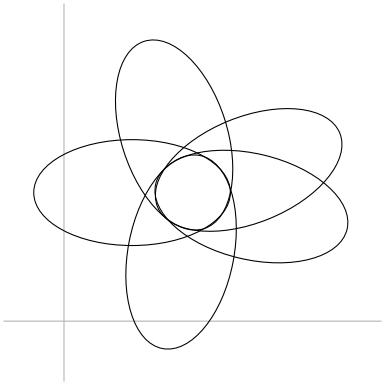


Figure 4.1: A simple example that illustrates the benefits of EP. The parameter $\boldsymbol{\theta}$ has two dimensions and data points have been partitioned into five pieces. Each ellipse represents an equi-contour of the likelihood corresponding to one partition. In the EP, the computational effort in the site inference can be focused into the area of overlap, whereas with independent parallel inferences, the whole ellipse has to be covered.

points, that is $K < N$, the inference in each individual unit might become slower and more informative. Because of this increased effort in the site calculations, parallelisating the computation becomes more effective, as the effort required in the information transfer between the nodes relative to the total computational effort in the parallel nodes decreases.

4.2 Implementation

This section presents a distributed parallel EP algorithm by Gelman et al. (2014a) using normal approximation. The posterior distribution

$$\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{y}) \propto \prod_{k=1}^{K} \mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta})\,\mathrm{p}(\boldsymbol{\theta}) \tag{4.1}$$

is approximated by

$$g(\boldsymbol{\theta}) \propto g_0(\boldsymbol{\theta}|\boldsymbol{\eta}_0, \boldsymbol{\Omega}_0) \prod_{k=1}^{K} g_k(\boldsymbol{\theta}|\boldsymbol{\eta}_k, \boldsymbol{\Omega}_k) = \mathrm{N}(\boldsymbol{\theta}|\boldsymbol{\eta}, \boldsymbol{\Omega}), \tag{4.2}$$

where all the site approximations $g_k(\boldsymbol{\theta}|\boldsymbol{\eta}_k, \boldsymbol{\Omega}_k)$ and prior $g_0(\boldsymbol{\theta}|\boldsymbol{\eta}_0, \boldsymbol{\Omega}_0)$ are normal. The term $g_0(\boldsymbol{\theta}|\boldsymbol{\eta}_0, \boldsymbol{\Omega}_0)$ corresponds to the prior $\mathrm{p}(\boldsymbol{\theta})$ directly or by approximating it with normal distribution. Normal distributions are written in the terms of the natural parameters of the exponential family:

$$\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1} \quad \text{and} \quad \boldsymbol{\eta} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \tag{4.3}$$

where $\boldsymbol{\Sigma}$ is the corresponding covariance matrix and $\boldsymbol{\mu}$ is the mean vector. Using these, a product of normal distributions can be calculated by summing the respective natural parameters:

$$\prod_{i=1}^{n} \mathrm{N}(\boldsymbol{x}\,|\,\boldsymbol{\eta}_i, \boldsymbol{\Omega}_i) = \mathrm{N}(\boldsymbol{x}\,|\,\boldsymbol{\eta}, \boldsymbol{\Omega}), \tag{4.4}$$

where

$$\boldsymbol{\Omega} = \sum_{i=1}^{n} \boldsymbol{\Omega}_i \quad \text{and} \quad \boldsymbol{\eta} = \sum_{i=1}^{n} \boldsymbol{\eta}_i. \tag{4.5}$$

In the initialisation of the algorithm, some starting values has to be assigned for the site approximations $g_k(\theta\,|\,\boldsymbol{\eta}_k, \boldsymbol{\Omega}_k)$. One possibility is to improperly set each site term to identity by setting $\boldsymbol{\Omega}_k = 0$ and $\boldsymbol{\eta}_k = 0$. In this case the initial global approximation corresponds to the prior, that is $g(\theta) = g_0(\theta\,|\,\boldsymbol{\eta}_0, \boldsymbol{\Omega}_0)$. Let $\Delta\boldsymbol{\Omega}_k$ and $\Delta\boldsymbol{\eta}_k$ denote the requested change in the site approximation k in the current iteration. Initially $\Delta\boldsymbol{\Omega}_k = 0$ and $\Delta\boldsymbol{\eta}_k = 0$, $\forall k \in 1, 2, \ldots, K$. The following algorithm is repeated until convergence:

1. Compute new updated site parameters with damping level $\delta \in (0, 1]$ for all $k = 1, 2, \ldots K$ by

$$\boldsymbol{\Omega}_k^{\text{new}} = \boldsymbol{\Omega}_k + \delta\Delta\boldsymbol{\Omega}_k, \qquad \boldsymbol{\eta}_k^{\text{new}} = \boldsymbol{\eta}_k + \delta\Delta\boldsymbol{\eta}_k. \tag{4.6}$$

2. Compute new natural parameters of $g(\boldsymbol{\theta})$ by

$$\boldsymbol{\Omega}^{\text{new}} = \boldsymbol{\Omega}_0 + \sum_{k=1}^{K} \boldsymbol{\Omega}_k^{\text{new}}, \qquad \boldsymbol{\eta}^{\text{new}} = \boldsymbol{\eta}_0 + \sum_{k=1}^{K} \boldsymbol{\eta}_k^{\text{new}}. \tag{4.7}$$

3. If $\boldsymbol{\Omega}^{\text{new}}$ is not positive definite, decrease δ and go back to step 1.

4. Form the cavity distributions $g_{-k}(\boldsymbol{\theta}) = \mathrm{N}(\boldsymbol{\theta}\,|\,\boldsymbol{\eta}_{-k}, \boldsymbol{\Omega}_{-k})$ for all $k = 1, 2, \ldots K$ by

$$\boldsymbol{\Omega}_{-k} = \boldsymbol{\Omega}^{\text{new}} - \boldsymbol{\Omega}_k^{\text{new}}, \qquad \boldsymbol{\eta}_{-k} = \boldsymbol{\eta}^{\text{new}} - \boldsymbol{\eta}_k^{\text{new}}. \tag{4.8}$$

5. If $\boldsymbol{\Omega}_{-k}$ is not positive definite for any k, decrease δ and go back to step 1. Otherwise, accept the new global approximation by setting $\boldsymbol{\Omega} = \boldsymbol{\Omega}^{\text{new}}$, $\boldsymbol{\eta} = \boldsymbol{\eta}^{\text{new}}$ and the site approximations by setting $\boldsymbol{\Omega}_k = \boldsymbol{\Omega}_k^{\text{new}}$, $\boldsymbol{\eta}_k = \boldsymbol{\eta}_k^{\text{new}}$ for all $k = 1, 2, \ldots K$.

6. Approximate the tilted distribution with a normal distribution $\mathrm{N}(\boldsymbol{\eta}_{\backslash k}, \boldsymbol{\Omega}_{\backslash k})$ and determine its natural parameters for all $k = 1, 2, \ldots K$. Various methods for performing this is discussed in section 3.2. Methods for approximating the precision matrix $\boldsymbol{\Omega}_{\backslash k}$ from MCMC samples is discussed in section 5.3. When the precision matrix has been estimated, $\boldsymbol{\eta}_{\backslash k}$ can be calculated from equation (4.5).

7. For all $k = 1, 2, \ldots K$ do the following: If the precision matrix $\boldsymbol{\Omega}_{\backslash k}$ was successfully estimated so that it is positive definite, compute the new site parameter updates $\Delta\boldsymbol{\Omega}_k$ and $\Delta\boldsymbol{\eta}_k$ so that the moments of the global approximation match with the tilted approximation and the cavity distribution according to equation (3.7), that is $\boldsymbol{\Omega}_k^{\text{new}} = \boldsymbol{\Omega}_{\backslash k} - \boldsymbol{\Omega}_{-k}$ and $\boldsymbol{\eta}_k^{\text{new}} = \boldsymbol{\eta}_{\backslash k} - \boldsymbol{\eta}_{-k}$ and particularly

$$\Delta\boldsymbol{\Omega}_k = \boldsymbol{\Omega}_{\backslash k} - \boldsymbol{\Omega}_{-k} - \boldsymbol{\Omega}_k, \qquad \Delta\boldsymbol{\eta}_k = \boldsymbol{\eta}_{\backslash k} - \boldsymbol{\eta}_{-k} - \boldsymbol{\eta}_k. \tag{4.9}$$

   If the precision matrix is not positive definite, the update can be discarded by setting $\Delta\boldsymbol{\Omega}_k = 0$ and $\Delta\boldsymbol{\eta}_k = 0$ or by forcing the precision matrix positive definite. This matter is discussed in more detail in section 5.2.

The steps 1–7 are repeated until the tilted distribution are consistent with the global approximation. The natural parameters $\boldsymbol{\Omega}_{\backslash k}$ and $\boldsymbol{\eta}_{\backslash k}$ should be close to the corresponding global approximation parameters $\boldsymbol{\Omega}$ and $\boldsymbol{\eta}$ for all $k = 1, 2, \ldots K$.

4.2.1 Parallelisation

The algorithm can be parallelised for optimal performance; after all, the algorithm is a parallel EP implementation. Consider a case where K site clients or nodes and a master node are available. Each site node is responsible of site specific calculations and the master node manages and combines information from the sites. Embarrassingly parallelisable steps include the steps 1, 4, 6 and 7. Determining the positive definiteness of the cavity precision matrix in the step 5 can be done in parallel already at the step 4. The steps 6 and 7 can be done in each site node in serial without contacting the master node in between. Thus one parallel EP iteration contains three parallel phases.

For the first parallelised phase covering the update of the site parameters, the master node needs to first contact each site node and wait for reply from them all. If the master node does not share memory with the site nodes, the updated site parameters need to be send to the master node in the reply. However, as the previous site parameters and site updates are already in the site nodes, only the damping factor needs to be passed to the site nodes when calling.

In the second parallel phase, the new candidate cavity distributions are calculated and checked for positive definiteness. Here the curent global approximation parameters needs to be passed to each site. For the reply, only information about the positive definiteness needs to be send back to the master node. Here the master node can also jump back to the step 1 prematurely, if any of the site nodes report failure in the positive definiteness. Otherwise, if all the site nodes report success, the master node can continue to the step 6.

The last parallel phase, where the site inference is conducted and the site parameter updates are calculated, is probably the most computationally heavy

part of the algorithm. The master node contacts each site node to indicate that the site parameter update was successful, but no other information needs to be passed in the calling message. Also the reply from the site nodes does not need to contain any other information that the indication of success in the estimation of the site parameter update terms. The next iteration can be started when all the sites have finished. However, here it is also possible for the master node to continue to the next iteration already when few of the site inferences have completed their task. This asynchronous parallelisation is discussed further in section 5.4.1.

4.3 Hierarchical setting

Using EP for distributed Bayesian inference provides significant benefits for hierarchical models. Consider a model that has J hierarchical groups with corresponding local parameter vectors $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_J$ and shared parameters $\boldsymbol{\phi}$. The shared parameters contains all the hyperparameters and other data model parameters, that are shared among the hierarchical groups. The information from the other sites' local parameters does not affect the inference of the others. Thus these parameters does not need to be distributed in the message passing information, that is the EP algorithm does not need to concern with them. The convergence of the EP has to happen only on the shared parameters.

Consider a partitioning of the data $\boldsymbol{y}$ into K sites $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K$, where each site k has its local model $\mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta}_k, \boldsymbol{\phi})\,\mathrm{p}(\boldsymbol{\theta}_k|\boldsymbol{\phi})$, that is each $\boldsymbol{\theta}_k$ affects only the site k. The EP can be applied here to approximate the marginal posterior distribution of the shared parameters $\mathrm{p}(\boldsymbol{\phi}|\boldsymbol{y})$. The algorithm follows the description in section 3.1, where the tilted distribution for site k is

$$g_{\backslash k}(\boldsymbol{\theta}_k, \boldsymbol{\phi}) \propto \mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta}_k, \boldsymbol{\phi})\,\mathrm{p}(\boldsymbol{\theta}_k|\boldsymbol{\phi})\,g_{-k}(\boldsymbol{\phi}) \tag{4.10}$$

and the approximation is made for its marginal distribution $g_{\backslash k}(\boldsymbol{\phi})$. After the final approximation for $\mathrm{p}(\boldsymbol{\phi}|\boldsymbol{y})$ is obtained from the EP, the joint posterior distribution can be approximated from

$$\mathrm{p}(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{y}) \propto \mathrm{p}(\boldsymbol{\phi}|\boldsymbol{y}) \prod_{k=1}^{K} \mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta}_k, \boldsymbol{\phi})\,\mathrm{p}(\boldsymbol{\theta}_k|\boldsymbol{\phi}), \tag{4.11}$$

where $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\right]$. This holds because each sites' local parameters $\boldsymbol{\theta}_k$ are assumed to be independent given the shared parameters $\boldsymbol{\phi}$. However, some additional considerations for approximating the joint posterior distribution is presented in section 4.3.1.

The computational gain of hierarchical partitioning in EP is probably big. Let N_J denote the number of samples and N_θ the number of local parameters per group (same for each group). Let N_ϕ denote the number of shared parameters.

The full problem without data partitioning has JN_J data points and $JN_\theta + N_\phi$ parameters but the distributed EP problem has K parallel problems with JN_J/K data points and $JN_\theta/K + N_\phi$ parameters. The scaling with respect to the data points occurs also when EP is applied to non-hierarchical problems, but the scaling with respect to the parameters applies especially for hierarchical problems. If J or N_θ is big, the benefit of the parallelisation is probably big, as quite often the computational complexity scales at least quadratically with respect to the number of parameters. For example, in the algorithm implemented in the experiments of this thesis, cubically scaling Cholesky decomposition is applied to the covariance matrix of the parameters.

The natural partitioning of the samples for utilising the full potential of the EP in hierarchical setting is to distribute all hierarchical groups to individual sites, that is $K = J$, where K is the number of sites and J is the number of groups. Then the approximation of the tilted distribution at each site simplifies to a non-hierarchical inference problem. In practice however, the number of hierarchical groups J is often considerable big and it might not be possible to effectively utilise as many parallel computing units. In those cases, it is possible to process many site in one computing unit in serial fashion. However, it might be beneficial to distribute multiple groups into single site instead, that is $K < J$, so that each site considers only $\lceil J/K \rceil$ or $\lfloor J/K \rfloor$ hierarchical groups.

Distributing the samples evenly by splitting the groups is probably not beneficial. This would increase the number of considered groups in the individual site inference problems and the corresponding local parameters should be included in the parameters distributed by the EP. By splitting the groups, the computational benefits of the hierarchical setting would be lost. If the number of available parallel computing units is greater than the number of hierarchical groups, when possible, it could be highly effective to introduce a nested EP algorithm instead. In this algorithm, the excess computing units are utilised by applying a higher level distributed EP into the inference of the tilted distribution inside the lower level EP. However, additional experimenting would be required in order to make more thorough comparison between conventional and nested EP in distributed setting for hierarchical models.

4.3.1 Approximating the joint posterior distribution

Approximating the joint posterior density after convergence of EP algorithm for a hierarchical problem requires some consideration. Usable factorisation for approximating the joint posterior is given in equation 4.11, where the marginal posterior of the shared parameters $\mathrm{p}(\boldsymbol{\phi}|\boldsymbol{y})$ can be substituted by the obtained EP approx-

imation $g(\boldsymbol{\phi})$:

$$g(\boldsymbol{\theta}, \boldsymbol{\phi}) = g(\boldsymbol{\phi}) \prod_{k=1}^{K} \mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta}_k, \boldsymbol{\phi}) \, \mathrm{p}(\boldsymbol{\theta}_k|\boldsymbol{\phi}) \,. \tag{4.12}$$

Different approaches for approximating this can be taken.

Consider here that sampling is used to approximate the tilted distributions in EP. If the marginal posterior distributions for each local parameter θ_k are required only separately, the samples obtained from the tilted distributions in the last iteration can be used directly as simulations of $\boldsymbol{\theta}_k$ or for approximating $\mathrm{p}(\boldsymbol{\theta}_k, \boldsymbol{\phi}|\boldsymbol{y})$. However, if the joint distribution of all the parameters or samples from it are required, the tilted distribution samples can not be used. Different site simulations are not synchronised so that different sites simulate different values of $\boldsymbol{\phi}$. One way of obtaining samples from the approximate joint posterior distribution is to first draw samples from the obtained EP approximation for the posterior $\mathrm{p}(\boldsymbol{\phi}|\boldsymbol{y})$ and, for each sample of $\boldsymbol{\phi}$, perform inference for each local parameter $\boldsymbol{\theta}_k$ conditional on $\boldsymbol{\phi}$ and draw a sample for each of them. Here the inferences for each $\boldsymbol{\theta}_k$ can be performed in parallel. Depending of the complexity of the conditional inferences, this method can be computationally very expensive, but may be enhanced for example by applying adiabatic Monte Carlo (Betancourt 2014).

4.4 Unknown hyperparameter

If an unknown prior parameter, that is a hyperparameter, is assigned for a parameter, it has to be specially dealt with. One possibility is to include the hyperparameter in the inferred parameters of the EP algorithm and assign a separate site for it. Consider a problem with parameter $\boldsymbol{\theta}$, that has a prior $\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{\gamma})$ with unknown hyperparameter $\boldsymbol{\gamma}$. The joint posterior distribution of the parameters is

$$\mathrm{p}(\boldsymbol{\theta}, \boldsymbol{\gamma}|\boldsymbol{y}) \propto \mathrm{p}(\boldsymbol{\theta}|\boldsymbol{\gamma}) \, \mathrm{p}(\boldsymbol{\gamma}) \prod_{k=1}^{K} \mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta}) \,. \tag{4.13}$$

In the EP algorithm described in section 3.1, the respective site terms $g_k(\boldsymbol{\theta}, \boldsymbol{\gamma})$ corresponds to usual likelihood terms $\mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta})$, where $k = 1, 2, \ldots, K$, and the hyperparameter site term $g_\gamma(\boldsymbol{\theta}, \boldsymbol{\gamma})$ to the term $\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{\gamma})$. The tilted distributions of the sites k are defined by

$$g_{\backslash k}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \propto \mathrm{p}(\boldsymbol{y}_k|\boldsymbol{\theta}) \, g_{-k}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \tag{4.14}$$

and the hyperparameter site tilted distribution by

$$g_{\backslash \gamma}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \propto \mathrm{p}(\boldsymbol{\theta}|\boldsymbol{\gamma}) \, g_{-\gamma}(\boldsymbol{\theta}, \boldsymbol{\gamma}), \tag{4.15}$$

where the term $p(\boldsymbol{\theta}|\boldsymbol{\gamma})$ is seen as a likelihood function of $\boldsymbol{\gamma}$. It can be seen that the inference subproblem for the hyperparameter site has different structure than the other sites and has to be handled differently. Notably the hyperparameter site does not consider the data $\boldsymbol{y}$ directly but through the cavity distribution.

Similarly as in the non-hierarchical case of EP described in section 4.1, if an unknown hyperparameter is assigned for a parameter in hierarchical model, it needs to be specially dealt with. However, if the prior affects only local parameters, it does not need to be included as a separate site. Only unknown priors of shared parameters require special attention. This is because the information for hyperparameter comes only from one site and the inference for it is done in the corresponding site itself. Anyhow, the hyperparameter of a local parameter itself, if it is shared among multiple groups, has to be included in the inferred parameters of the EP algorithm, just as described in the start of section 4.3.

As mentioned before in section 4.3, splitting a hierarchical group into multiple sites requires that corresponding local parameters are included in the parameters inferred by EP. If such a parameter has an unknown hyperparameter, also the additional site corresponding to that hyperparameter has to be included in the EP sites, just as for hyperparameters of a global parameter. From EP point of view, a local parameter is not local anymore, if it is split into multiple sites.

Chapter 5

Algorithmic considerations

This chapter discusses some implementational aspects of using EP for distributed Bayesian inference as described in chapter 4. These considerations do not concern only the implementation in section 4.2, but are more generally applicable. Section 5.1 discusses the convergence of the EP algorithm by considering it as a stochastic optimisation problem. Section 5.2 addresses with the problem of encountering a covariance and precision matrix, that is not positive definite, and failing tilted distribution approximations. Different techniques for obtaining better tilted distribution precision matrix approximations from samples is discussed in section 5.3. Finally some additional implementational considerations and possible enhancements are presented in section 5.4.

5.1 Stochastic optimisation

When considering the algorithm descibed in section 4.2, it can be seen that it is a stochastic optimisation algorithm with similarities to a Robbins–Monro algorithm (Robbins and Monro 1951). The damping factor corresponds to the step size and the moment update terms described in the step 7 corresponds to the parameter updates. In order for such algorithm to converge succesfully into right solution, several requirements must be met. In the context of EP, two criteria should be payed attention to: the damping factor has to decrease and the tilted distribution moment approximations has to be unbiased.

The decreasing nature of the damping factor can be ensured by controlling the initial damping factor for example by decreasing it exponentially. If the damping factor is constantly too big, the noise in the moments may cause the algorithm to oscillate near the solution without ever reaching it. The forceful decreasing of the damping factor has to be done even if the tilted distributon moments could be calculated precisely. It is also even more important in parallel EP, as multiple update steps are performed simultaeously.

If the positive definiteness check decreases the damping factor into a small

value, the EP algorithm has either converged to a local or global mode or the noise in the moment estimates is too big. In such a case, it could be beneficial to estimate the moment update terms againg from the sites, possibly with increased accuracy, in order to check if the latter has occurred. However, detecting the convergence to a local mode is harder. Here it should be noted that, if the moment estimates are precise enough, the EP algorithm can converge even if the damping factor is big.

In general, the tilted distributon moment approximations in the described EP algorithm are biased. This may affect the convergence properties of the algorithm either by increasing the approximation error or by preventing the convergence completely. Thus it is necessary to pay attention to the accuracy of the estimates. Different variance reduction methods and increased number of samples can be used to reduce the risk of failure in the convergence. More detailed discussion about estimating the precision matrix from samples is presented in section 5.3.

5.2 Constraining positive definiteness

The EP algorithm described in section 4.2 contains three checks for positive definiteness of a covariance or precision matrix, which ensure that the computations remain well defined at all times:

- global posterior approximation at step 3 after updating the site approximations,
- cavity distributions at step 5,
- tilted distribution approximation at step 7.

The check for the global approximation is not always necessary however, as ensuring that all of the covariance matrices of the cavity distributions are positive definite also ensures that the global approximation is well defined. The check for the global approximation can be used as a shortcut for detecting a bad case before more numerically heavy step 4 with all the site nodes is reached. Also if the moments of the global approximation are monitored during the iterations, this check comes for free when inverting the precision matrix with Cholesky decomposition.

The algorithm definition in section 4.2 uses damping to cope with situations where the global approximation or the cavity distribution covariance matrices are not positive definite. The damping controls how much of the update in the site approximations determined in the previous iteration are taken into account in current iteration. The damping is first set to some value in the range $(0, 1]$ and is then decreased towards zero until all of the cavity distributions pass the positive definiteness check. If the EP algorithm result oscillates without ever

converging, damping can be used to forcefully decay the updates by assigning a starting damping factor that decreases as a function of the iterations. A starting damping factor is the initial level of damping in each iteration. In the beginning of the EP algorithm, the starting damping factor should be high in order to support fast convergence.

The check for the tilted distribution covariance matrix is related to the approximation method. For example, if the Laplace's method or its derivatives are used, the accuracy of the found mode and its neighbourhood affect the outcome of the covariance matrix. For sample based MCMC methods, the number of samples affect the accuracy considerably. If the tilted distribution approximation fails, that is the resulting covariance matrix estimate is not positive definite, various alternative actions can be taken. The simplest one is to discard the update and hope that sufficient amount of change from the other sites are provided in order to have successful approximation on the next iteration. Another solution is to force the variance to positive or the covariance matrix to positive definite.

Forcing the variance of a one dimensional parameter to positive is trivial. However, forcing a precision or covariance matrix $\boldsymbol{A}$ to positive definite can be done in various ways. Here three alternative methods are presented in the order of computational complexity (most complex first):

Eigendecomposition

Perform an eigendecomposition for the matrix $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$. Keep the eigenvectors $\boldsymbol{Q}$ but replace any negative eigenvalues in $\boldsymbol{\Lambda}$ with small positive number, and reconstruct the matrix.

Eigenvalue shift

Find the smallest eigenvalue of the matrix $\boldsymbol{A}$ (which is non-positive as $\boldsymbol{A}$ is not positive definite) and add its absolute value and a small buffer value to the diagonal entries of $\boldsymbol{A}$. Eigenvalues λ and corresponding eigenvectors $\boldsymbol{x}$ of matrix $\boldsymbol{A}$ satisfy the equation $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. The eigenvalue equation of a new matrix $\boldsymbol{A} + c\mathbf{I}$, where $c \in \mathbb{R}$, is

$$(\boldsymbol{A} + c\mathbf{I})\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + c\mathbf{I}\boldsymbol{x} = \lambda\boldsymbol{x} + c\boldsymbol{x} = (\lambda + c)\boldsymbol{x}, \tag{5.1}$$

that is every eigenvalue of $\boldsymbol{A}$ is shifted by c. If c is greater than the smallest eigenvalue of $\boldsymbol{A}$, the matrix is positive definite.

Gershgorin circle shift

Replace all the diagonal elements, that are smaller than the sum of the absolute values of the other elements in the corresponding row, by a value slightly greater than the sum. This ensures positive definiteness because according to the Gershgorin circle theorem, the eigenvalues of a matrix $\boldsymbol{A}$

live in balls centred at the diagonal entries $[\boldsymbol{A}]_{i,i}$ with radius equal to the described row sum $\sum_{j:j\neq i}\left|[\boldsymbol{A}]_{i,j}\right|$.

The eigendecomposition method has been used in similar setting for example by Betancourt (2013). This method is computationally heavy for big matrices and may introduce significant numerical error from the decomposition that accumulate due to the reconstruction of the matrix in the end. With this method however, the analytical change in the eigenvalues of the modified matrix is minimal and the eigenvectors are preserved.

Finding and approximating the smallest eigenvalue of a big matrix is a lot easier and more accurate than finding them all with eigendecomposition. The smallest eigenvalue can be approximated efficiently for example by Lanczos iteration with selective reorthogonalisation (Parlett and Scott 1979). Because of this, the eigenvalue shift method is numerically faster and may introduce less error than the eigendecomposition method. On the other hand, when compared to the eigendecomposition method, shifting all the eigenvalues and particularly shifting all the diagonal elements of an estimated covariance or precision matrix may be considered to be too big of a change. However, often the estimated matrix is only slightly deviated from a positive-definite matrix, so that its smallest eigenvalue is close to zero. In such a case, the change in the diagonal is small compared to the greater numerical error introduced by the eigendecomposition method. Similarly as in the eigendecomposition method, the eigenvectors of the matrix are preserved in the eigenvalue shift method.

Computationally simplest method is the Gershgorin circle shift method. Similarly as the eigenvalue shift method, it also changes all the eigenvalues of the matrix. However, unlike with the eigenvalue shift method, it also changes the eigenvectors and not much can be said about the magnitude of the change in the eigenvalues. On the other hand, it preserves all the diagonal elements of the matrix that satisfy the row sum criteria. Both methods preserve all the off diagonal elements.

If an approximation of the tilted distribution precision matrix has failed, it would be sensible to assume that the accuracy of the estimate is quite low already in the first place. With this in mind, it seems justified to use the easiest method to force the matrix positive definite. Thus the Gershgorin circle shift method described earlier would probably do just fine. If more accuracy is required or the eigenvectors should be preserved, the eigenvalue shift method should be used and preferred over the eigendecomposition method. If the minimum eigenvalue is found to be a large negative number, eigendecomposition should be used instead. In such a case, however, the original matrix is quite erroneous and discarding the update completely would probably be the best choice. Further analysis and experimenting would be required in order to make more rigorous conclusions about

the effect of the error in practice.

5.3 Precision matrix estimation

Estimating the moments of the tilted distribution in an EP algorithm is not easy. The problem was first discussed in a general level in section 3.2. In many cases, the problematic estimation target is the population covariance matrix $\boldsymbol{\Sigma}$ of d dimensional random variable $\boldsymbol{\theta}$, that is

$$\boldsymbol{\Sigma} = \mathrm{Var}(\boldsymbol{\theta}) = \mathrm{E}\Big((\boldsymbol{\theta} - \boldsymbol{\mu})(\boldsymbol{\theta} - \boldsymbol{\mu})^{\mathrm{T}}\Big), \tag{5.2}$$

where $\boldsymbol{\mu} = \mathrm{E}(\boldsymbol{\theta})$. This section considers the case, where the precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ is required and should be estimated from n samples $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n$ of $\boldsymbol{\theta}$.

The naive way of estimating $\boldsymbol{\Omega}$ is to first estimate $\boldsymbol{\Sigma}$ and then invert it. The $d \times d$ scatter matrix $\boldsymbol{S}$ is defined as

$$\boldsymbol{S} = \sum_{i=1}^{n} \Big(\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}\Big)\Big(\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}\Big)^{\mathrm{T}} = \Big(\boldsymbol{\Theta} - \bar{\boldsymbol{\theta}}\mathbb{1}_n^{\mathrm{T}}\Big)\Big(\boldsymbol{\Theta} - \bar{\boldsymbol{\theta}}\mathbb{1}_n^{\mathrm{T}}\Big)^{\mathrm{T}}, \tag{5.3}$$

where $\bar{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\theta}_i$ is the sample mean and $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \ldots & \boldsymbol{\theta}_n \end{bmatrix}$ is the $d \times n$ sample data matrix obtained by concatenating all the sample vectors column-wise. The natural estimator for $\boldsymbol{\Sigma}$ is the sample covariance matrix

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\boldsymbol{S}. \tag{5.4}$$

The sample precision matrix is the inverse of this matrix

$$\widehat{\boldsymbol{\Omega}} = \widehat{\boldsymbol{\Sigma}}^{-1} = (n-1)\boldsymbol{S}^{-1}. \tag{5.5}$$

Both $\widehat{\boldsymbol{\Sigma}}$ and $\bar{\boldsymbol{\theta}}$ are unbiased estimators regardless of the underlying true distribution, that is $\mathrm{E}\Big(\widehat{\boldsymbol{\Sigma}}\Big) = \boldsymbol{\Sigma}$ and $\mathrm{E}\Big(\bar{\boldsymbol{\theta}}\Big) = \boldsymbol{\mu}$. However, in general $\widehat{\boldsymbol{\Omega}}$ is biased, that is $\mathrm{E}\Big(\widehat{\boldsymbol{\Omega}}\Big) \neq \boldsymbol{\Omega}$. Moreover, different maximum likelihood estimates for $\boldsymbol{\Sigma}$ and unbiased estimators for $\boldsymbol{\Omega}$ can be derived for certain distributions. As discussed in the section 5.1, it would be highly beneficial to be able to obtain an unbiased estimate of the precision matrix of the tilted distribution from MCMC samples. However, the experiments in chapter 6 shows that it is possible to obtain good results also using biased estimates.

Despite being an unbiased estimator, the sample covariance matrix is not a good estimate for the eigenvalues of the true covariance matrix. Thus its inverse is a poor estimate for $\boldsymbol{\Omega}$. If the number of samples is much greater than the number of dimensions, that is in the case of standard asymptotics (Le Cam and Yang 2000), the sample precision matrix is sufficiently accurate. However, when n is

bigger than d but still relatively close to it, the sampling error of $\widehat{\boldsymbol{\Sigma}}$ is significantly big (Bai and Shi 2011). In addition, according to the theorem A.2, the rank of $\widehat{\boldsymbol{\Sigma}}$ can not be greater than n. Thus, if $n < d$, the sample precision matrix is not invertible. In this extreme case one would have to resort for example to a least squares solution with Moore–Penrose pseudoinverse.

If the true distribution of $\boldsymbol{\theta}$ is from the normal family, the expectation of the inverse of the sample covariance matrix is

$$\mathrm{E}\big(\widehat{\boldsymbol{\Sigma}}^{-1}\big) = \frac{n-1}{n-d-2}\boldsymbol{\Omega}. \tag{5.6}$$

Thus the unbiased sample precision matrix estimator can be constructed by multiplying out the biased factor (Muirhead 2005, p. 136):

$$\widehat{\boldsymbol{\Omega}}_{\mathrm{N}} = \frac{n-d-2}{n-1}\widehat{\boldsymbol{\Sigma}}^{-1} = (n-d-2)\boldsymbol{S}^{-1}. \tag{5.7}$$

Furthermore, other improved sample based estimates with smaller risk can be derived for normal distribution (Tsukuma and Konno 2006) and for some other more general distribution families (Bodnar and Gupta 2011; Gupta, Varga and Bodnar 2013; Sarr and Gupta 2009).

In a general case, where no assumptions can not be made about the distribution of $\boldsymbol{\theta}$, different propositions for improved precision matrix estimates has been made. These methods mainly utilise two different approaches: shrinking and sparsifying. The former shrinks the eigenvalues of the covariance matrix and the latter imposes sparse constrains to its structure. These methods are further discussed in sections 5.3.1 and 5.3.2 respectively.

Most of the covariance and precision matrix estimators are quite sensitive to outliers. In addition to shrinkage and sparse estimators, a number of robust methods for dealing with outliers directly has been proposed. Minimum covariance determinant estimator is one such a method (Rousseeuw 1984; Rousseeuw and Driessen 1999). It uses only a portion of samples neglecting outliers and rescales and reweights the estimate. Using these methods for MCMC simulations is not beneficial, however, as the samples should not contain outliers in the first place.

5.3.1 Shrinkage estimators

As the problem with using the inverse of the sample covariance matrix as the precision matrix estimate is that the eigenvalues are not accurate, a good idea for improvement would be to control them somehow. Shrinkage estimators forms a linear combination of the sample estimator with some target. Depending on the method, the target matrix can be for example the identity matrix or heuristically chosen prior matrix. With some wise choice of coefficients, the procedure

shrinks the eigenvalues of the sample covariance matrix, thus the name shrinkage estimator.

The shrinkage method was first introduced by Stein (1956). Ledoit and Wolf (2004) later proposed an extended method called linear shrinkage estimator by applying identity matrix as the target. They showed that the results are well-behaved and optimal in the quadratic mean sense. More recently, they further extended the idea into nonlinear shrinkage estimator by utilising the theory of the random matrices (Ledoit and Wolf 2012).

Optimal linear shrinkage estimator (OLSE), proposed by Bodnar, Gupta and Parolya (2014a), is a generalisation of the linear shrinkage estimator, where the target matrix can be an arbitrary deliberate positive-definite symmetric matrix. The estimator forms the covariance matrix which still needs to be inverted in order to get the precision matrix. Bodnar, Gupta and Parolya (2014b) later extended the method to directly estimate the precision matrix. The OLSE estimator for the precision matrix is given by

$$\widehat{\boldsymbol{\Omega}}_{\mathrm{OLSE}} = \widehat{\alpha}\widehat{\boldsymbol{\Omega}} + \widehat{\beta}\boldsymbol{\Omega}_0, \tag{5.8}$$

where

$$\widehat{\boldsymbol{\Omega}} = n\boldsymbol{S}^{-1}, \tag{5.9}$$

$$\widehat{\alpha} = 1 - \frac{d}{n} - \frac{\frac{1}{n}\left\|\widehat{\boldsymbol{\Omega}}\right\|_{\mathrm{tr}}^2 \left\|\boldsymbol{\Omega}_0\right\|_{\mathrm{F}}^2}{\left\|\widehat{\boldsymbol{\Omega}}\right\|_{\mathrm{F}}^2 \left\|\boldsymbol{\Omega}_0\right\|_{\mathrm{F}}^2 - \mathrm{tr}\left(\widehat{\boldsymbol{\Omega}}\boldsymbol{\Omega}_0\right)^2} \tag{5.10}$$

and

$$\widehat{\beta} = \left(1 - \frac{d}{n} - \widehat{\alpha}\right)\frac{\mathrm{tr}\left(\widehat{\boldsymbol{\Omega}}\boldsymbol{\Omega}_0\right)}{\left\|\boldsymbol{\Omega}_0\right\|_{\mathrm{F}}^2}. \tag{5.11}$$

The *oracle* OLSE estimator, where $\boldsymbol{\Omega}_0$ is the true unknown precision matrix, is a consistent estimator for the precision matrix under high-dimensional asymptotics (Bodnar, Gupta and Parolya 2014b). The *bona fide* OLSE estimator, where a chosen prior precision matrix is used as $\boldsymbol{\Omega}_0$, is optimal in the sense of the Frobenius loss. Providing relevant prior information on the spectrum of the precision matrix into $\boldsymbol{\Omega}_0$ can significantly improve the estimator. If no relevant prior information is available, one naive choice is to set $\boldsymbol{\Omega}_0 = \mathbf{I}/n$. In the context of the tilted distribution approximation in the EP algorithm, the cavity distribution $g_{-i}(\theta)$ is convenient and likely effective choice for the prior.

5.3.2 Sparse estimators

Sparse precision matrix estimators assumes that either the covariance or the precision matrix or both are sparse matrices. Here it is noteworthy that, in general,

the inverse of a sparse matrix is not sparse. If the matrix is block diagonal or can be permuted into such, the inverse is also sparse. This requires that $\boldsymbol{\theta}$ can be partitioned into sets that are independent of each other. In other words, an element in the precision matrix is zero, if the corresponding variables are independent conditionally on the others. Such assumptions are often reasonable and yields better conditioned estimates when justified.

Possibly the most popular sparse precision matrix estimator is the graphical lasso method (Friedman, Hastie and Tibshirani 2008). It uses L1 penalty to enforce sparsity on the precision matrix. The parameter for controlling the sparsity level can be selected using cross-validation. Various improvements and modifications for this method has been proposed (Witten, Friedman and Simon 2011; Zhang and Zou 2014).

Sparse precision matrix estimators tend to work better than shrinkage based ones when the number of samples n is smaller than the number of variables d. However, because sparse precision matrix estimators try to learn independences from the data, they are numerically unstable with highly correlated data. Also if the number of samples n is much greater than the number of variables d, the shrinkage based estimators tend to work better. Figure 5.1 shows comparisons of sample, OLSE and graphical lasso estimates for simulated sparse and full precision matrices. Clearly the graphical lasso method works best for the former case and OLSE for the latter case.

5.3.3 Control variates

In addition to the tailored covariance estimator methods, some general MC variance reduction methods might also be usable in the estimation of the covariance or precision matrix from samples. Control variates is one such a technique. It was first introduced by Boyle (1977) in a context of financial option price estimation. It uses auxiliary estimates of known quantities to counter the error in the evaluation of the target estimate. Applying the control variates for moment estimation is described in appendix B.

In the estimation of the tilted distribution moments in the EP algorithm, control variates can be naturally constructed by using the corresponding cavity distribution as the approximative distribution with known expectation. While control variates method is often applicable and convenient, using it for estimating moments from MCMC samples, is problematic. This is because the probability density of the target distribution at the samples should be known in normalised form and estimating the required normalisation constant is in general a hard task.

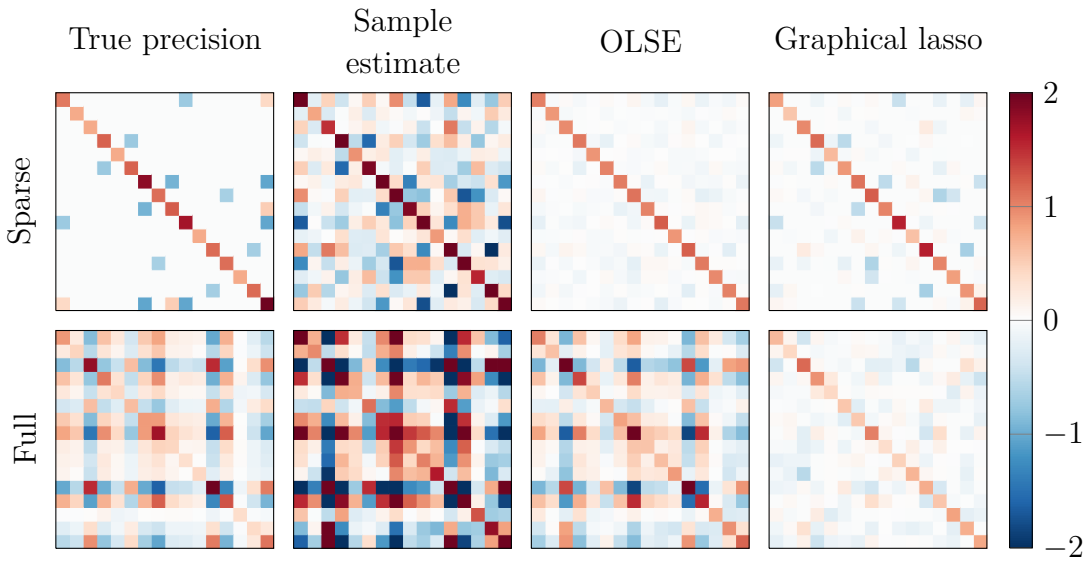


Figure 5.1: Comparison of different precision matrix estimators for simulated data sets. In the first row, the true underlying covariance has sparse structure, and in the second row, the true covariance matrix is full. The data set contains 32 samples drawn from a corresponding normal distribution with 16 dimensions. For the sparse case, the sparsity parameter of the graphical lasso estimator has been selected with cross-validation. In the full covariance case however, the sparsity has been set manually to a low value, as cross-validation results in zero sparsity, which corresponds to the sample estimate.

5.4 Other considerations

The following considerations apply for different algorithmic situations related to using EP for distributed Bayesian inference. All of them propose some small changes or additional features to the EP algorithm introduced in chapter 4 that aims to make the inference computationally more effective or stable.

5.4.1 Asynchronous parallelisation

As pointed out before in section 4.1, when multiple points are distributed to one site in EP, the inference on the site becomes harder and slower. In such a setting, and especially if the difference in the number of points in each site is large, some inference units might complete their computation sooner than others. In these cases, it could be beneficial to update the global approximation in the master node as soon as two or more nodes has finished the inference. These faster nodes could then start a new iteration with updated information while other nodes are still busy. Different techniques for determining when to prematurely combine sites can be designed and applied depending on the problem.

5.4.2 First iteration estimate

As the effectiveness of the distributed EP method comes from the message passing after the first iteration, it could be beneficial to use the first iteration of the EP algorithm to get only a rough estimate of the likelihood contributions for the sites. If sample based methods are used to perform the inference on the tilted distribution, the speed-up can be achieved for example by simply decreasing the number of obtained samples.

5.4.3 Mixing the samples

Conventionally the final posterior approximation for $\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{y})$ in EP is obtained from the current global approximation $g(\boldsymbol{\theta})$. If sampling is used to approximate the tilted distributions in the sites, another option is to form the final approximation by mixing the samples from the last iteration site approximations. Mixing the samples is justified, because after the EP algorithm has converged, the tilted distributions should be consistent with each other and with the global approximation. Because this sample based approximation resembles the EP-based marginal improvements described by Cseke and Heskes (2011), mixing the samples could take possible skewness in the posterior distribution better into account.

5.4.4 Reusing simulations

If the tilted distributions in EP are approximated using MCMC methods, the last sample of the previous iteration serves as a convenient starting point for the next iteration simulation. In addition, other samples could also be reused from the previous iterations with importance sampling. Several approaches for this importance sampling scheme exists. A basic method is given for example by Barthelmé and Chopin (2014) and more elaborate one by Cornuet et al. (2012). Reusing samples from the previous iterations will likely be very effective especially when the EP algorithm is close to convergence. At the convergence, the samples should be obtained practically for free.

5.4.5 Smoothing

Another way of countering oscillation in the EP is to apply smoothing into the tilted distributions; in every iteration, the approximated tilted distribution is shifted into the direction of the previous tilted distributions. Smoothing is more general form of damping, where only the previous iteration tilted distribution is considered. In smoothing, arbitrary many previous tilted distributions can be considered, though intuitively it would be sensible to weight more resent iterations more than the older ones.

Shifting the distributions can be done in many ways. If sample based estimates are used to approximate the first and second moment of the tilted distribution, combining the samples from the previous iterations seems reasonable. Instead of storing the samples $\boldsymbol{\theta}$ from each iteration, an efficient way of conducting this sample combining is to store the scatter matrix $\boldsymbol{S}_i$ presented in equation 5.3, the sample mean $\bar{\boldsymbol{\theta}}_i$ and the number of contributing samples n_i from sufficiently many previous iterations $i = 1, 2, \ldots, p$ and combine them accordingly:

$$n^{\text{new}} = \sum_{i=1}^{p} w_i n_i, \tag{5.12}$$

$$\bar{\theta}^{\,\text{new}} = \frac{1}{n^{\text{new}}} \sum_{i=1}^{p} w_i n_i \bar{\theta}_i, \tag{5.13}$$

$$\boldsymbol{S}^{\text{new}} = \sum_{i=1}^{p} w_i \Big(\boldsymbol{S}_i + n_i \big(\bar{\theta}_i - \bar{\theta}^{\,\text{new}}\big)\big(\bar{\theta}_i - \bar{\theta}^{\,\text{new}}\big)^{\mathrm{T}} \Big), \tag{5.14}$$

where w_i is the desired weight for the corresponding iteration.

Chapter 6

Experiments

This chapter presents the experiments conducted for testing the EP algorithm discussed in chapter 4. Multiple simulated hierarchical linear regression and classification problems are constructed and tested with different setup. All the problems have $J = 100$ hierarchical groups and $D = 20$ dimensional explanatory variable. The data sets are simulated from the models with 40–60 samples per group. The EP algorithm is tested with $K = 2, 4, 8, 16, 32, 64$ sites.

The models used in the experiments are defined in section 6.1. Sections 6.2 and 6.3 describes how the data sets are simulated. The EP algorithm and the algorithmic choices related to it are discussed in section 6.4. Finally, in section 6.5, the results are presented.

6.1 Model definitions

In these experiments, the EP algorithm for distributed inference is tested with several hierarchical linear regression and classification problems. In both of these, four different model structures are defined, and for each of these, uncorrelated and correlated explanatory variable structures are used to generate the data. The model structures are denoted by #1 – #4.

In the following model definitions, $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_D \end{bmatrix}^{\mathrm{T}}$ denotes the explanatory variable vector with D components and y denotes the response variable. Model parameter α is the intercept coefficient and $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_D \end{bmatrix}^{\mathrm{T}}$ is the regression coefficient vector.

6.1.1 Problem types

The experimented problems in this thesis are of two different main types: linear regression and classification. The general form of the linear regression experiment

models is:

$$y|\boldsymbol{x},\alpha,\boldsymbol{\beta},\sigma \sim \mathrm{N}\big(f(\boldsymbol{x}),\sigma\big), \tag{6.1}$$

$$\sigma \sim \text{log-N}(0,\sigma_0) \tag{6.2}$$

and similarly the general form of the classification models is:

$$y|\boldsymbol{x},\alpha,\boldsymbol{\beta} \sim \text{Bernoulli}\Big(\text{logit}^{-1}\big(f(\boldsymbol{x})\big)\Big), \tag{6.3}$$

where the latent variable $f(\boldsymbol{x})$ is defined in general level as

$$f(\boldsymbol{x}) = \alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}. \tag{6.4}$$

Function $\text{logit}^{-1}(t)$ is the inverse logit function or the logistic function given by

$$\text{logit}^{-1}(t) = \frac{\mathrm{e}^t}{\mathrm{e}^t+1} = \frac{1}{1+\mathrm{e}^{-t}}. \tag{6.5}$$

The precise definition of the latent variable and related parameters depends on the hierarchical model structure. However, as the linear regression problem contains the parameter σ, that is not directly related to the latent variable $f(\boldsymbol{x})$, it has to be added into the shared parameters $\boldsymbol{\phi}$ defined in the hierarchical model structures. As seen from equation (6.3), classification problems does not have such a parameter.

6.1.2 Hierarchical models

Four different models for the latent variable $f(\boldsymbol{x})$ are defined. Each of these impose different hierarchical structure to the resulting model. In the following, $j = 1,2,\ldots,J$ denotes the index of the associated hierarchical group of the parameters and $d = 1,2,\ldots,D$ denotes the index of the explanatory variable. The defined model structures are:

Model #1:

$$\begin{aligned} f(\boldsymbol{x}) &= \alpha_j + \boldsymbol{\beta}\boldsymbol{x} \\ \alpha_j &\sim \mathrm{N}(0,\sigma_\alpha) \\ \beta_d &\sim \mathrm{N}(0,\sigma_\beta) \\ \sigma_\alpha &\sim \text{log-N}\Big(0,\sigma^{\text{hyper}}_{\sigma,\alpha}\Big) \\ \boldsymbol{\phi} &= \Big[\log(\sigma_\alpha) \quad \boldsymbol{\beta}\Big]^{\mathrm{T}}, \end{aligned}$$

Model #2:

$$\begin{aligned} f(\boldsymbol{x}) &= \alpha_j + \boldsymbol{\beta}_j\boldsymbol{x} \\ \alpha_j &\sim \mathrm{N}(0,\sigma_\alpha) \\ [\boldsymbol{\beta}_j]_d &\sim \mathrm{N}\Big(0,[\boldsymbol{\sigma}_\beta]_d\Big) \\ \sigma_\alpha &\sim \text{log-N}\Big(0,\sigma^{\text{hyper}}_{\sigma,\alpha}\Big) \\ [\boldsymbol{\sigma}_\beta]_d &\sim \text{log-N}\Big(0,\Big[\boldsymbol{\sigma}^{\text{hyper}}_{\boldsymbol{\sigma},\beta}\Big]_d\Big) \\ \boldsymbol{\phi} &= \Big[\log(\sigma_\alpha) \quad \log(\boldsymbol{\sigma}_\beta)\Big]^{\mathrm{T}}, \end{aligned}$$

Model #3:

$$
\begin{aligned}
f(\boldsymbol{x}) &= \alpha_j + \boldsymbol{\beta}_j \boldsymbol{x} \\
\alpha_j &\sim \mathrm{N}(\mu_\alpha, \sigma_\alpha) \\
[\boldsymbol{\beta}_j]_d &\sim \mathrm{N}\Big([\boldsymbol{\mu}_\beta]_d, [\boldsymbol{\sigma}_\beta]_d\Big) \\
\mu_\alpha &\sim \mathrm{N}\Big(0, \sigma^{\text{hyper}}_{\mu,\alpha}\Big) \\
[\boldsymbol{\mu}_\beta]_d &\sim \mathrm{N}\Big(0, \Big[\boldsymbol{\sigma}^{\text{hyper}}_{\boldsymbol{\mu},\boldsymbol{\beta}}\Big]_d\Big) \\
\sigma_\alpha &\sim \text{log-N}\Big(0, \sigma^{\text{hyper}}_{\sigma,\alpha}\Big) \\
[\boldsymbol{\sigma}_\beta]_d &\sim \text{log-N}\Big(0, \Big[\boldsymbol{\sigma}^{\text{hyper}}_{\boldsymbol{\sigma},\boldsymbol{\beta}}\Big]_d\Big) \\
\boldsymbol{\phi} &= \Big[\mu_\alpha \quad \log(\sigma_\alpha) \quad \boldsymbol{\mu}_\beta \quad \log(\boldsymbol{\sigma}_\beta)\Big]^{\mathrm{T}},
\end{aligned}
$$

Model #4:

$$
\begin{aligned}
f(\boldsymbol{x}) &= \alpha_j + \boldsymbol{\beta}_j \boldsymbol{x} \\
\alpha_j &\sim \text{Laplace}(\mu_\alpha, \sigma_\alpha) \\
[\boldsymbol{\beta}_j]_d &\sim \text{Laplace}\Big([\boldsymbol{\mu}_\beta]_d, [\boldsymbol{\sigma}_\beta]_d\Big) \\
\mu_\alpha &\sim \text{Laplace}\Big(0, \sigma^{\text{hyper}}_{\mu,\alpha}\Big) \\
[\boldsymbol{\mu}_\beta]_d &\sim \text{Laplace}\Big(0, \Big[\boldsymbol{\sigma}^{\text{hyper}}_{\boldsymbol{\mu},\boldsymbol{\beta}}\Big]_d\Big) \\
\sigma_\alpha &\sim \text{half-Cauchy}\Big(0, \sigma^{\text{hyper}}_{\sigma,\alpha}\Big) \\
[\boldsymbol{\sigma}_\beta]_d &\sim \text{half-Cauchy}\Big(0, \Big[\boldsymbol{\sigma}^{\text{hyper}}_{\boldsymbol{\sigma},\boldsymbol{\beta}}\Big]_d\Big) \\
\boldsymbol{\phi} &= \Big[\mu_\alpha \quad \log(\sigma_\alpha) \quad \boldsymbol{\mu}_\beta \quad \log(\boldsymbol{\sigma}_\beta)\Big]^{\mathrm{T}}.
\end{aligned}
$$

In the hierarchical model definitions, the vector $\boldsymbol{\phi}$ denotes the shared parameters inferred with the EP algorithm. However, if used with linear regression base model, the base error variance $\log(\sigma)$ has to be added into it. The positive constrained variance parameters added to the vector $\boldsymbol{\phi}$ are first transformed to unconstrained space, so that a joint normal distribution can be used to approximate the posterior of $\boldsymbol{\phi}$ better.

6.2 Simulating data sets

Simulating data sets from the models is straightforward. First suitable parameters are set or sampled based on the hyperparameters. Then the explanatory variable is sampled from some selected distribution. Finally the response variable is determined by calculating $f(\boldsymbol{x})$ and sampling from the corresponding distribution.

The simulated data set are build using $J = 100$ hierarchical groups with each containing a random number of samples between 40 and 60. In our realisation, the resulting total number of simulated data points is 4979 in all the problems.

The number of explanatory variables is set to $D = 20$. The parameters of the models in every problem, if present, are selected in the following manner:

- Linear regression noise variance $\sigma^2 = 1$. The prior variance for the noise σ_0 is not ordered as σ is selected manually.
- Group variance of the intercept of the latent variable $\sigma_\alpha^2 = 1$. The hyperparameter $\sigma_{\sigma,\alpha}^{\text{hyper}}$ is not ordered as σ_α is selected manually.
- Group mean of the intercept of the latent variable $\mu_\alpha = 0.1$. The hyperparameter $\sigma_{\mu,\alpha}^{\text{hyper}}$ is not ordered as μ_α is selected manually.
- The hyperparameter of the group mean of a coefficient of the latent variable $\left[\boldsymbol{\sigma}_{\boldsymbol{\mu},\boldsymbol{\beta}}^{\text{hyper}}\right]_d^2 = 1$, $\forall d = 1, 2, \ldots, D$.
- The hyperparameter of the group variance of a coefficient of the latent variable $\left[\boldsymbol{\sigma}_{\boldsymbol{\sigma},\boldsymbol{\beta}}^{\text{hyper}}\right]_d^2 = 1$, $\forall d = 1, 2, \ldots, D$.

The explanatory variable is sampled from normal distribution: $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$, where all the components have the same mean and they are homoscedastic in every problem. In linear regression problems, the parameter $\boldsymbol{\mu}_{\boldsymbol{x}}$ is set to zero, as the mean does not affect the uncertainty in the outcome of the data set. In the classification problems however, it has to be regulated and it is set homogeneously to $\boldsymbol{\mu}_{\boldsymbol{x}} = \mu_{\boldsymbol{x}}\mathbb{1}$. For the covariance parameter $\boldsymbol{\Sigma}_{\boldsymbol{x}}$, two different types of structures are used:

- uncorrelated explanatory variable: $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2\mathbf{I}$,
- correlated explanatory variable: $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2\boldsymbol{\Sigma}_0$.

where the controlled parameter is $\sigma_{\boldsymbol{x}}^2$. More detailed description on the uncertainty is presented in section 6.3.

In the definition of the correlated explanatory variable covariance, the base covariance structure $\boldsymbol{\Sigma}_0$ is a random covariance matrix created using modified vines method described by Lewandowski, Kurowicka and Joe (2009). Their example implementation generates random correlation matrices that are uniformly distributed in the appropriate subset of $\mathbb{R}^{D(D-1)/2}$. In general, such correlation matrices have small off-diagonal elements when the number of variables is big. In these experiments however, a modified version, as suggested by Lewandowski et al., is used to generate correlation matrices with larger correlations; the partial correlations are sampled from Beta$(2, 2)$ linearly transformed to range $[-0.8, 0.8]$. Using this method, a random correlation matrix is sampled and assigned into the covariance structure $\boldsymbol{\Sigma}_0$ directly. Thus $\boldsymbol{\Sigma}_0$ is normalised to have unit variance in each variable.

6.3 Regulating the uncertainty

When simulating data sets from the models, it is necessary to pay attention to the difficulty of fitting it. The resulting data set should not be too easy to fit into the model but the uncertainty in the data should not be too high either. The resulting uncertainty can be measured and controlled in many ways. In these experiments, suitable explanatory variable distribution parameters $\mu_{\boldsymbol{x}}$ and $\sigma^2_{\boldsymbol{x}}$ are determined conditional on the model parameters separately for each hierarchical group. Naturally, this requires that the explanatory variable is sampled separately for each group. An alternative way would have been to determine suitable values without conditioning on the model parameters and use common sampling distribution of x for each group. In the following two sections, denoting the conditioning on the model parameters and the hierarchical group index is omitted for the sake of clarity.

6.3.1 Linear regression

In linear regression experiments, the uncertainty is controlled by analysing how much of the variability in the response variable y comes from the explanatory variable and how much from the noise. The ration of these two should be constant between different experiments.

Let random variable $F = f(\boldsymbol{x}) = \alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}$. The controlled measure is the coefficient of determination

$$R^2 = [\mathrm{Cor}(y, F)]^2 = \frac{\mathrm{Var}(F)}{\mathrm{Var}(y)}. \tag{6.6}$$

The above equation is derived in the appendix C. The variances of F and y are determined by the model parameters $\boldsymbol{\beta}$ and σ and by the covariance matrix of the explanatory variable $\boldsymbol{x}$:

$$\mathrm{Var}(F) = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{\beta} \qquad \mathrm{Var}(y) = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{\beta} + \sigma^2. \tag{6.7}$$

For emphasis, let $R^2_0 \in (0, 1)$ denote a selected target value for R^2. Substituting the variances in equations (6.7) into equation (6.6) gives

$$\frac{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{\beta}}{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{\beta} + \sigma^2} = R^2_0, \tag{6.8}$$

$$\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{\beta} = \frac{R^2_0}{1 - R^2_0}\sigma^2. \tag{6.9}$$

It can be seen from equations (6.6) and (6.7) that the location parameter $\boldsymbol{\mu}_{\boldsymbol{x}}$ and intercept coefficient α does not affect the coefficient of determination. Thus it is sufficient to regulate only the covariance structure $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ while $\boldsymbol{\mu}_{\boldsymbol{x}}$ is set to zero.

Substituting the covariances in equation (6.9) with the defined structures $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2\mathbf{I}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2\boldsymbol{\Sigma}_0$ gives the direct rules for determining the controlled parameter $\sigma_{\boldsymbol{x}}^2$ in the case of uncorrelated and correlated explanatory variable respectively:

$$\sigma_{\boldsymbol{x}}^2 = \frac{R_0^2}{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}(1-R_0^2)}\sigma^2 \qquad \text{or} \qquad \sigma_{\boldsymbol{x}}^2 = \frac{R_0^2}{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{\beta}(1-R_0^2)}\sigma^2. \tag{6.10}$$

It can be seen from these equations that smaller elements in the vector $\boldsymbol{\beta}$ results in bigger value for the covariance scale $\sigma_{\boldsymbol{x}}^2$. For numerical stability, the realisations of $\boldsymbol{\beta}$ are regulated so that $\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}$ is forced to be greater than $1 \cdot 10^{-4}$ by using rejection sampling.

6.3.2 Classification

In classification experiments, the uncertainty is controlled by setting restrictions to the random variable $P = \mathrm{logit}^{-1}\big(f(\boldsymbol{x})\big)$, that is the probability of y falling into one of the classes. The aim is to control the distribution of P so that it rarely gets values near zero or one. In such a case, the class of y is not too certain but dependent on the noise. The controlling of P is done by first selecting the model parameters and then assigning suitable parameters for the distribution of $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$.

In the following, we inspect the effect of the distribution of $\boldsymbol{x}$ into P conditional on the model parameters and present restrictions to the tail probabilities of P. Similarly as in section 6.3.1, let random variable $F = f(\boldsymbol{x}) = \alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}$. Because F is a sum of normally distributed random variables, its distribution is also normal and

$$\mathrm{E}(F) = \alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{x}}, \qquad \mathrm{Var}(F) = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{\beta}. \tag{6.11}$$

The distribution of P is logit-normal with $P \sim \text{logit-N}\big(\mathrm{E}(F), \mathrm{Var}(F)\big)$ and its cumulative distribution function is (Hinde 2014)

$$\Pr(P \leq p) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mathrm{logit}(p) - \mathrm{E}(F)}{\sqrt{2\,\mathrm{Var}(F)}}\right)\right]. \tag{6.12}$$

The function $\mathrm{erf}(t) = \frac{2}{\sqrt{\pi}}\int_0^{\pi} \mathrm{e}^{-t^2}\,\mathrm{d}t$ is the Gaussian error function, which is strictly increasing and its inverse can be approximated in various ways (Strecok 1968). The restrictions are assigned so that

$$\Pr(P \leq p_0) < \gamma_0 \quad \text{and} \quad \Pr(P > p_0) < \gamma_0, \tag{6.13}$$

where p_0 and γ_0 are both some small probability values bellow 0.5. When denoting both of the above inequations with $\Pr(P \leq p) \lesseqgtr \gamma$, where $<$ applies when $(p, \gamma) =$

(p_0, γ_0) and $>$ when $(p, \gamma) = (1 - p_0, 1 - \gamma_0)$, we get

$$\operatorname{erf}\left(\frac{\operatorname{logit}(p) - \operatorname{E}(F)}{\sqrt{2\operatorname{Var}(F)}}\right) \lessgtr 2\gamma - 1. \tag{6.14}$$

Because $\operatorname{erf}(t)$ is strictly increasing, its inverse can be applied so that

$$\frac{\operatorname{logit}(p) - \operatorname{E}(F)}{\sqrt{2\operatorname{Var}(F)}} \lessgtr \operatorname{erf}^{-1}(2\gamma - 1). \tag{6.15}$$

As both p_0 and γ_0 are assumed to be smaller than 0.5, both terms $\operatorname{logit}(p_0)$ and $\operatorname{erf}^{-1}(2\gamma_0 - 1)$ are negative. Furthermore, because $\operatorname{logit}(t) = -\operatorname{logit}(1 - t)$ and $\operatorname{erf}^{-1}(t) = -\operatorname{erf}^{-1}(-t)$, the pair of inequations (6.15) can be combined into one sufficient inequation:

$$\sqrt{\operatorname{Var}(F)} < \frac{\operatorname{logit}(p_0) + \left|\operatorname{E}(F)\right|}{\sqrt{2}\,\operatorname{erf}^{-1}(2\gamma_0 - 1)}. \tag{6.16}$$

In order for a solution to exist, it is clear that the right side of this inequation must be positive. This gives gives us a necessary condition

$$\left|\operatorname{E}(F)\right| < -\operatorname{logit}(p_0). \tag{6.17}$$

However, it is also necessary to add some clearance to this condition by selecting the smallest acceptable standard deviation for F denoted by $\sigma_{f,0}$ and defining an upper bound $\delta_{f,\max}$ for the magnitude of the mean of F by

$$\left|\operatorname{E}(F)\right| \leq \delta_{f,\max} \coloneqq \sqrt{2}\,\sigma_{f,0}\operatorname{erf}^{-1}(2\gamma_0 - 1) - \operatorname{logit}(p_0). \tag{6.18}$$

It can be seen from equation (6.16), that the standard deviation of $f(\boldsymbol{x})$ reaches its maximum value

$$\sqrt{\operatorname{Var}(F)} \leq \sigma_{f,\max} \coloneqq \frac{\operatorname{logit}(p_0)}{\sqrt{2}\,\operatorname{erf}^{-1}(2\gamma_0 - 1)} \tag{6.19}$$

when $\operatorname{E}(F) = 0$. Here it should also be noted that, in order for the condition (6.18) to be satisfiable, the threshold value $\sigma_{f,0}$ has to be selected so that it is smaller than $\sigma_{f,\max}$. The threshold values used in the experiments of this thesis are presented in the table 6.1.

The derived conditions can be used to select suitable values for the parameters of $\boldsymbol{x}$ in various ways. The following describes the selected procedure in these experiments. First it is inspected, whether or not the mean condition (6.18) is satisfied when $\boldsymbol{\mu}_{\boldsymbol{x}} = 0$, that is check if $|\alpha| \leq \delta_{f,\max}$. If this condition is satisfied, $\boldsymbol{\mu}_{\boldsymbol{x}}$ is set to zero and $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ is set so that the combined mean variance condition (6.16) is satisfied precisely. In the other case, $\boldsymbol{\mu}_{\boldsymbol{x}}$ is set so that the mean condition (6.18)

Table 6.1: The threshold values for restricting the tail probabilities of the classification probability P. The selected values p_0 and γ_0 are utilised in equation (6.13) and $\sigma_{f,0}$ in equation (6.18). The resulting value $\delta_{f,\max}$ is defined in equation (6.18) and $\sigma_{f,\max}$ in equation (6.19). The resulting upper bounds are presented approximately.

Selected values			Resulting upper bounds	
p_0	γ_0	$\sigma_{f,0}$	$\delta_{f,\max}$	$\sigma_{f,\max}$
0.2	0.01	0.25	0.80	0.60

is satisfied precisely with smallest possible change in the mean of F and $\boldsymbol{\Sigma_x}$ is set so that $\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma_x}\boldsymbol{\beta} = \sigma_{f,0}^2$.

With this explanatory variable parameter selection method, the distribution of P does not depend on the values of $\boldsymbol{\beta}$ or the dimensionality D. Parameter α affects the distribution by tilting it towards zero or one so that

$$P \sim \text{logit-N}\left(\alpha', \frac{\text{logit}(p_0) + |\alpha'|}{\sqrt{2}\,\text{erf}^{-1}(2\gamma_0 - 1)}\right), \tag{6.20}$$

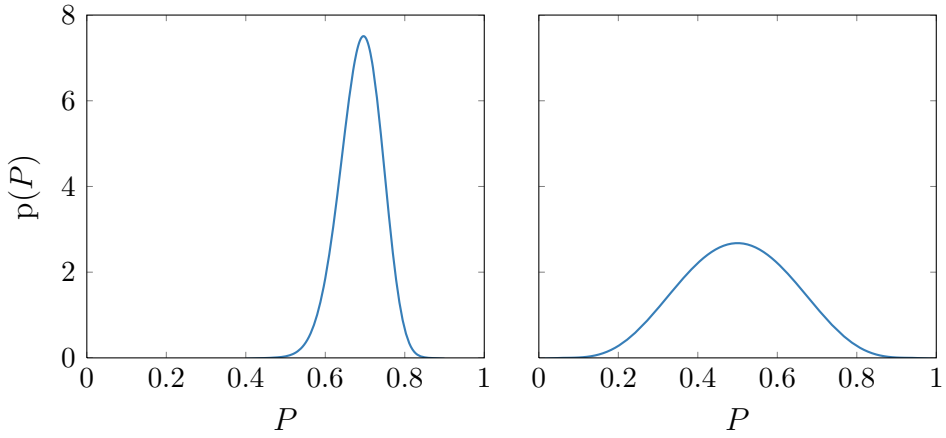
where

$$\alpha' = \min\big(\max(\alpha, -\delta_{f,\max}), \delta_{f,\max}\big). \tag{6.21}$$

The equation (6.21) limits the effect of α to the range $[-\delta_{f,\max}, \delta_{f,\max}]$, where the maximally tilted distributions $P \sim \text{logit-N}\big(\pm\delta_{f,\max}, \sigma_{f,0}\big)$ corresponds to the border values. As mentioned before, the maximal variance case $P \sim \text{logit-N}\big(0, \sigma_{f,max}\big)$ is achieved, when $\alpha = 0$. Figure 6.1 illustrates these two extreme distributions corresponding to the used parameter values shown in the table 6.1.

In the definition, the mean parameter is restricted to $\boldsymbol{\mu_x} = \mu_{\boldsymbol{x}}\mathbb{1}$ and the covariance parameter either to $\boldsymbol{\Sigma_x} = \sigma_{\boldsymbol{x}}^2\mathbf{I}$ or to $\boldsymbol{\Sigma_x} = \sigma_{\boldsymbol{x}}^2\boldsymbol{\Sigma}_0$ corresponding to the cases of uncorrelated and correlated explanatory variables respectively. Regardless of the chosen explanatory variable structure, applying equation (6.18) for the mean parameter $\mu_{\boldsymbol{x}}$ simplifies to

$$\mu_{\boldsymbol{x}} = \begin{cases} \dfrac{\delta_{f,\max} - \alpha}{\sum_{i=1}^{D}\beta_i}, & \text{if } \alpha > \delta_{f,\max}, \\ \dfrac{-\delta_{f,\max} - \alpha}{\sum_{i=1}^{D}\beta_i}, & \text{if } \alpha < -\delta_{f,\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.22}$$



(a) $P \sim \text{logit-N}\big(+\delta_{f,\max}, \sigma_{f,0}\big)$ (b) $P \sim \text{logit-N}\big(0, \sigma_{f,max}\big)$

Figure 6.1: Illustration of the resulting extreme distributions of the classification probability P using the selected parameter values presented in the table 6.1. Subfigure 6.1a corresponds to the case of $|\alpha| \geq \delta_{f,\max}$ and subfigure 6.1b to the case of $\alpha = 0$. These distributions do not depend on the parameter $\boldsymbol{\beta}$.

In the case $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2 \mathbf{I}$, the variance parameter $\sigma_{\boldsymbol{x}}$ becomes

$$\sigma_{\boldsymbol{x}} = \begin{cases} \dfrac{\text{logit}(p_0) + |\alpha|}{\text{erf}^{-1}(2\gamma_0 - 1)\sqrt{2\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}}}, & \text{if } |\alpha| \leq \delta_{f,\max}, \\ \dfrac{\sigma_{f,0}}{\sqrt{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}}}, & \text{otherwise,} \end{cases} \tag{6.23}$$

and in the case $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2 \boldsymbol{\Sigma}_0$

$$\sigma_{\boldsymbol{x}} = \begin{cases} \dfrac{\text{logit}(p_0) + |\alpha|}{\text{erf}^{-1}(2\gamma_0 - 1)\sqrt{2\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{\beta}}}, & \text{if } |\alpha| \leq \delta_{f,\max}, \\ \dfrac{\sigma_{f,0}}{\sqrt{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{\beta}}}, & \text{otherwise.} \end{cases} \tag{6.24}$$

6.4 Methods

The hierarchical EP algorithm presented in chapter 4 and particularly in section 4.3 is experimented with the problems defined in sections 6.1 to 6.3 using various settings. The algorithm is implemented in Python programming language and it uses various packages and externalisations for numerical efficiency. More detailed technical description and the program itself is available online (Sivula 2014). The implementation uses parallel EP but the inference on the sites is not parallelised. The site inference and the full reference models are conducted using Stan probabilistic programming language (Stan Development Team 2014b) with PyStan interface (Stan Development Team 2014a). The programs are run using

computer resources of the Science-IT project within the Aalto University School of Science.

The implemented EP algorithm approximates the joint posterior distribution of the shared parameters $\boldsymbol{\phi}$ by a normal distribution. The setting in the EP algorithm are the following:

- The number of sites is set to $K = 2, 4, 8, 16, 32, 64$.
- The tilted distribution precision matrix is approximated with the sample estimator, OLSE or graphical lasso estimator with cross-validation. The OLSE estimator uses the known precision matrix of the current cavity distribution as the prior for the estimated precision matrix. The graphical lasso (denoted with G-lasso-CV) is implemented in Scikit-learn library for Python (Pedregosa et al. 2011).
- Damping is applied by using exponentially decreasing damping factor with decay rate 0.22 (multiplied by 0.8 in each site update failure). The algorithm stops if the damping factor decreases below $1 \cdot 10^{-6}$. The initial damping factor in each iteration also decays exponentially as a function of the iteration. In the first iteration, it is set to 1 and it decreases towards zero with decay rate 0.18.
- Site inference is conducted with a no-U-turn Hamiltonian MCMC sampler with 8 chains and 500 iterations, of which 250 is used as a warm-up. These settings result in 2000 samples. The effect of the number of samples is tested by conducting selected experiments also with half the chains resulting in 1000 samples. The last samples from the MCMC iteration are used as a starting point for the inferences in the next EP iterations.
- The prior for the shared parameters is $\boldsymbol{\phi} \sim \mathrm{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, where $\boldsymbol{\mu}_0 = 0$ and $\boldsymbol{\Sigma}_0 = 1.5^2 \cdot \mathbf{I}$.
- If the tilted distribution precision matrix estimation fails, that is the estimated matrix is not positive definite, the site update is set to zero. However, the used number of samples in the estimation is relatively high and it is unlikely that the estimation fails.

The corresponding full model using the whole data set without EP or partitioning is used for comparison. The normal approximation for the posterior of the full model for each problem is formed by sampling from the same Stan model used for corresponding tilted distributions. The settings for the sampling of the full model are set to four chains, 15000 samples per chain, 5000 samples burn-in and every second sample discarded, which result in a total of 20000 samples.

Each EP algorithm is run for 26 iterations or until the damping factor reaches the threshold. In every iteration, the current approximation for the posterior of

θ, denoted by $\mathrm{N}_{\mathrm{EP}} = \mathrm{N}(\boldsymbol{\mu}_{\mathrm{EP}}, \boldsymbol{\Sigma}_{\mathrm{EP}})$, is compared against the normal posterior approximation from the full model, denoted by $\mathrm{N}_{\mathrm{full}} = \mathrm{N}(\boldsymbol{\mu}_{\mathrm{full}}, \boldsymbol{\Sigma}_{\mathrm{full}})$. Two different error measures are used: mean squared error (MSE) and KL-divergence. The MSE is calculated between the means:

$$\mathrm{MSE}(\boldsymbol{\mu}_{\mathrm{full}} \| \boldsymbol{\mu}_{\mathrm{EP}}) = \frac{1}{D} \sum_{d=1}^{D} ([\boldsymbol{\mu}_{\mathrm{EP}}]_d - [\boldsymbol{\mu}_{\mathrm{full}}]_d)^2, \tag{6.25}$$

and the KL-divergence is calculated from the EP approximation to full approximation (Kullback 1959, p. 189):

$$\begin{aligned} \mathrm{KL}(\mathrm{N}_{\mathrm{full}} \| \mathrm{N}_{\mathrm{EP}}) = \frac{1}{2} \Bigg(& (\boldsymbol{\mu}_{\mathrm{EP}} - \boldsymbol{\mu}_{\mathrm{full}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathrm{EP}}^{-1} (\boldsymbol{\mu}_{\mathrm{EP}} - \boldsymbol{\mu}_{\mathrm{full}}) \\ & + \mathrm{tr}\Big(\boldsymbol{\Sigma}_{\mathrm{EP}}^{-1} \boldsymbol{\Sigma}_{\mathrm{full}}\Big) - D + \log\left(\frac{|\boldsymbol{\Sigma}_{\mathrm{EP}}|}{|\boldsymbol{\Sigma}_{\mathrm{full}}|}\right) \Bigg). \end{aligned} \tag{6.26}$$

6.5 Results

This section presents the results of the experiments. As stated before, multiple simulated hierarchical linear regression and classification problems are constructed and tested with different setup. All the problems have $J = 100$ hierarchical groups and $D = 20$ dimensional explanatory variable. The data sets are simulated from the models with 40–60 samples per group. The EP algorithm is tested with $K = 2, 4, 8, 16, 32, 64$ sites.

The results of the experiments are shown in the figures 6.2–6.9. Figure 6.2 shows that the average step size of the MCMC iterations are bigger when the data set is partitioned into smaller sets. This supports the intuition, that the site inference becomes faster and easier when the number of partitions is increased.

The effect of the used number of sites and different variance reduction methods are shown in the figure 6.3. It can be seen from this figure, that increasing the number of partitions increases the approximation error. Using different variance reduction methods may improve the results in some situations, but may also introduce bias and thus increase the error in the outcome.

Figures 6.4 and 6.5 shows that the approximation error decreases when the EP algorithm advances and all the examples converge in 26 iterations. The MSE error does not behave as smoothly as the KL-divergence error because it only considers the means of the approximation.

Figures 6.6 and 6.7 compares the mean and standard deviation of individual shared parameters between the EP and full approximation for $K = 16$. The means have little difference but the standard deviations, particularly for $\boldsymbol{\sigma}_\beta$, have a systematic difference.

The effect of the precision of the tilted distribution moment estimates is illustrated in the figure 6.8. The error in the resulting approximation is smaller with more precise estimate in every example. The settings for the damping factor are the same for all the experiments. It can also be seen that the difference is relatively big, when the number of sites and the magnitude of the error is big.

Finally, in the figure 6.9, the difference in the KL-divergence with correlated and uncorrelated explanatory variable is shown. It can be seen from this figure that the number of sites has different effect on the resulting approximation error on different data sets; the data set with uncorrelated explanatory variable results in a smaller approximation error than the data set with correlated explanatory variable when $K = 2$, but the trends bypasses each other between $K = 32$ and $K = 64$.

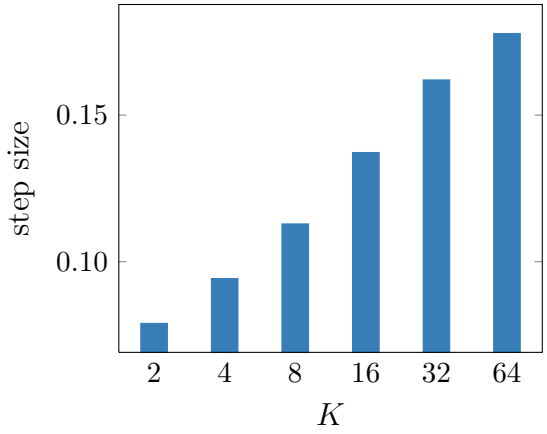


Figure 6.2: The average step size in the MCMC iterations in all the problems with different number of groups. In this example, the explanatory variable is correlated and sample estimate is used to approximate tilted distribution moments.

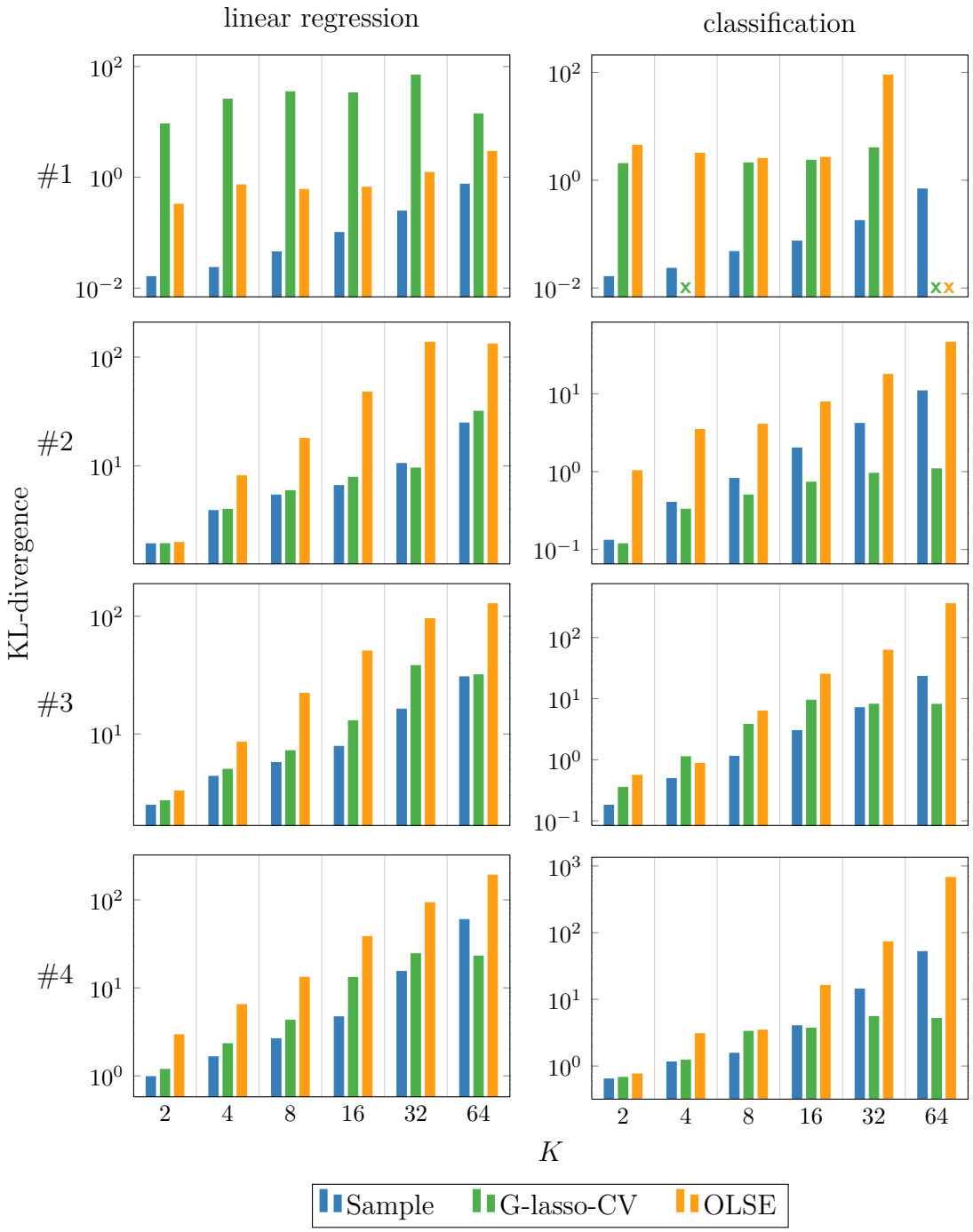


Figure 6.3: Final obtained KL-divergence with correlated explanatory variable with different tilted distribution moment estimates. In the upper right graph, **x** marks failed runs that did not converge. Columns correspond to linear regression and classification problems correspondingly and rows corresponds to different model structures. The y-axis is in the logarithmic scale.

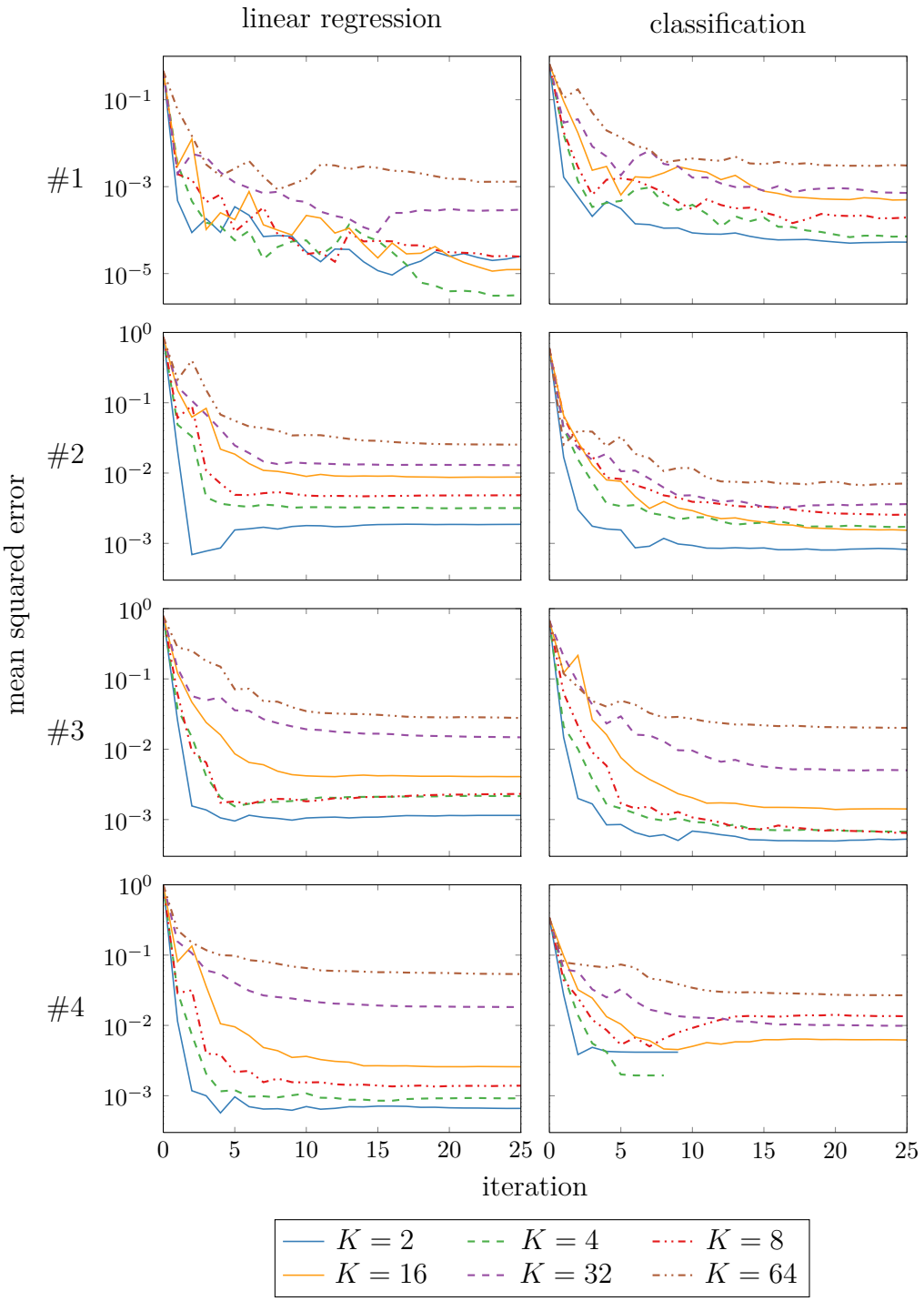


Figure 6.4: MSE for each iteration with correlated explanatory variable and sample estimate. Colours denote the used number of sites. In the lower right graph, lines corresponding to $K = 2$ and $K = 4$ end prematurely because the damping factor reaches the threshold. Columns correspond to linear regression and classification problems correspondingly and rows corresponds to different model structures. The y-axis is in the logarithmic scale.

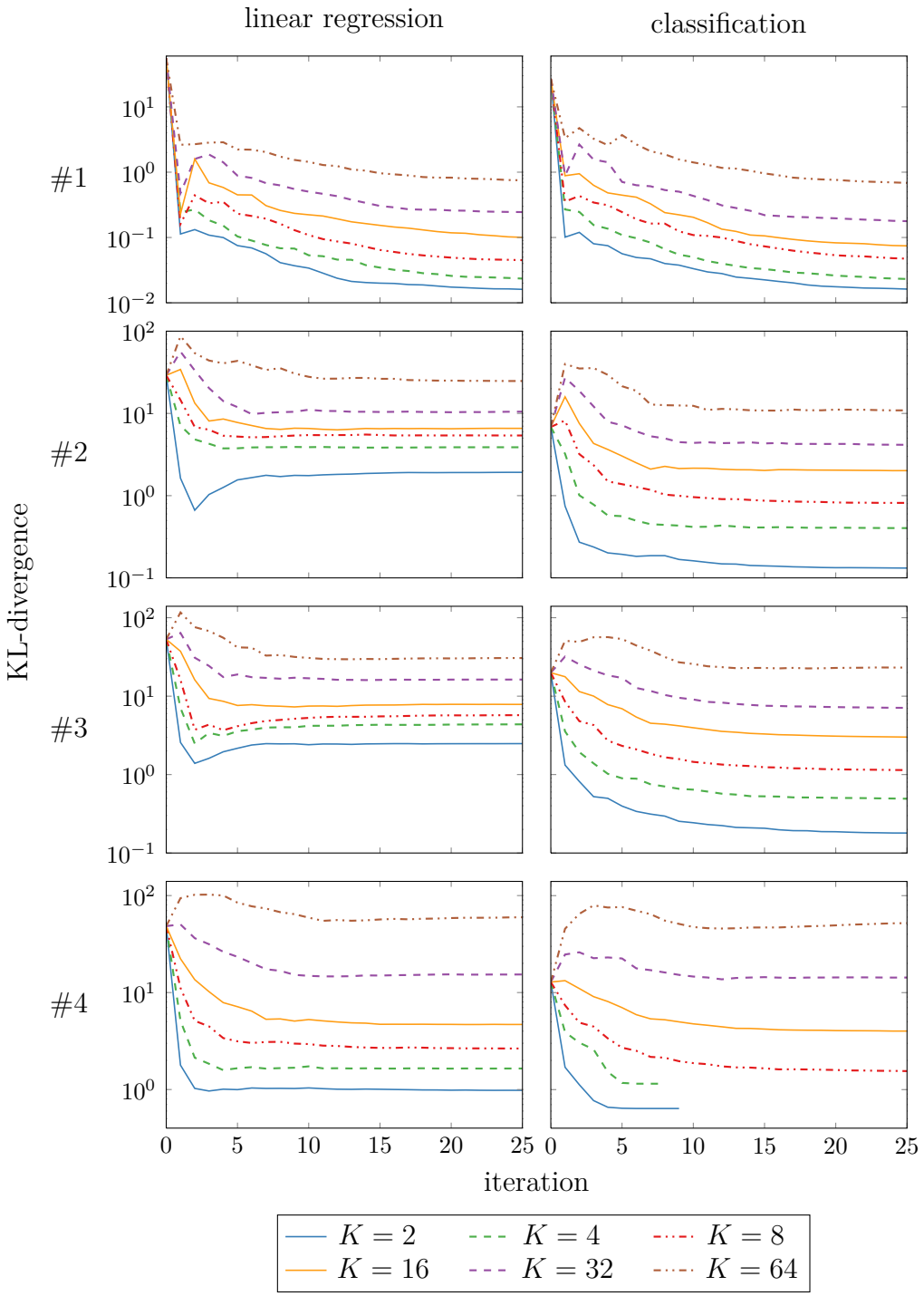


Figure 6.5: KL-divergence for each iteration with correlated explanatory variable and sample estimate. Colours denote the used number of sites. In the lower right graph, lines corresponding to $K = 2$ and $K = 4$ end prematurely because the damping factor reaches the threshold. Columns correspond to linear regression and classification problems correspondingly and rows corresponds to different model structures. The y-axis is in the logarithmic scale.

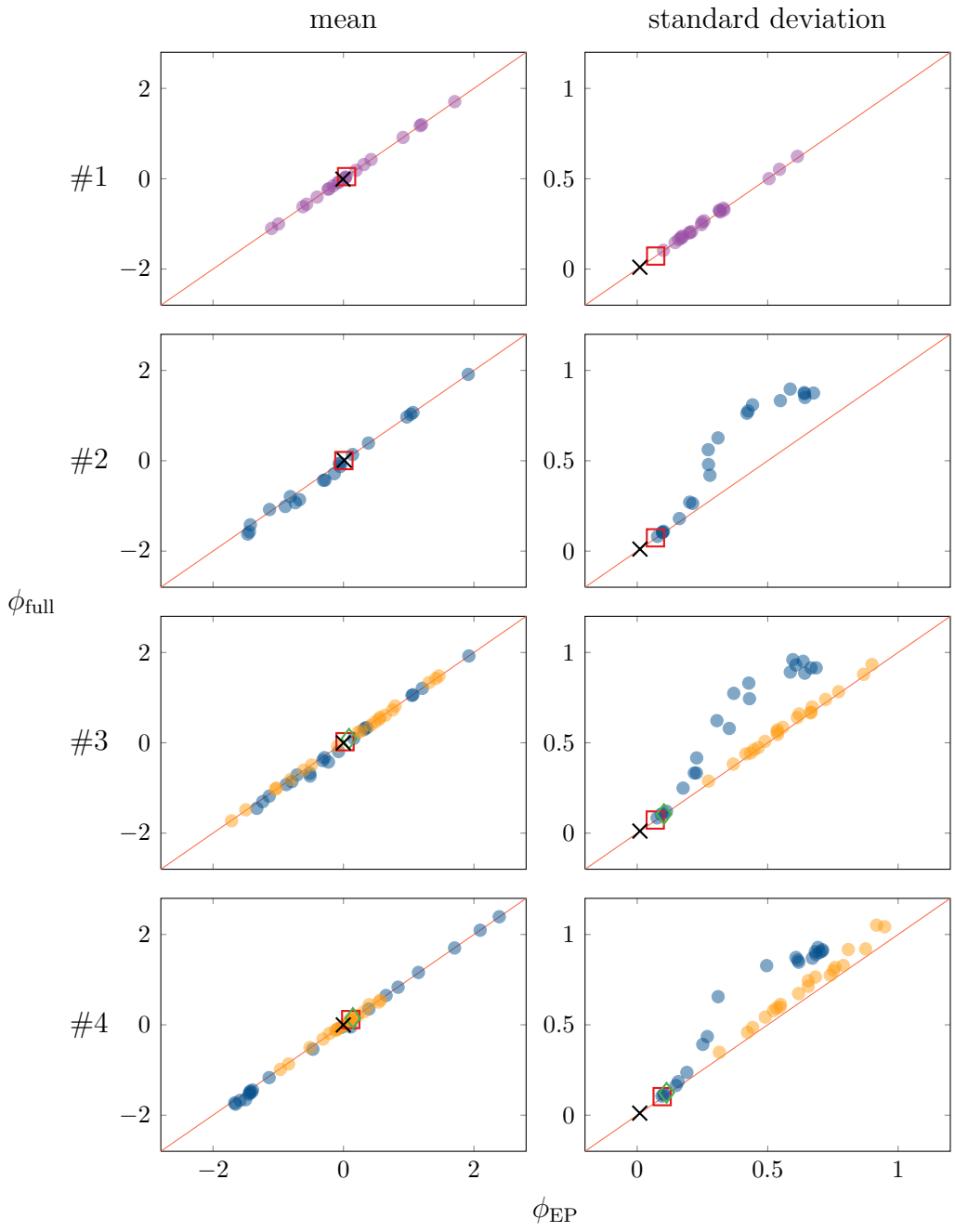


Figure 6.6: Comparison of the posterior mean and standard deviation of the shared parameters between the EP (ϕ_{EP}) and the full (ϕ_{full}) approximation in the linear regression problem. In this example, the number of sites $K = 16$, the explanatory variable is correlated and sample estimate is used to approximate tilted distribution moments. The resulting EP approximation has been constructed by mixing the samples from all the sites. Markers and the corresponding parameters are: ◊ μ_α, □ $\log(\sigma_\alpha)$, ● $\boldsymbol{\mu}_\beta$, ● $\log(\boldsymbol{\sigma}_\beta)$, ● $\boldsymbol{\beta}$ and × σ. The red diagonal line shows the points of equivalence. Rows corresponds to different model structures.

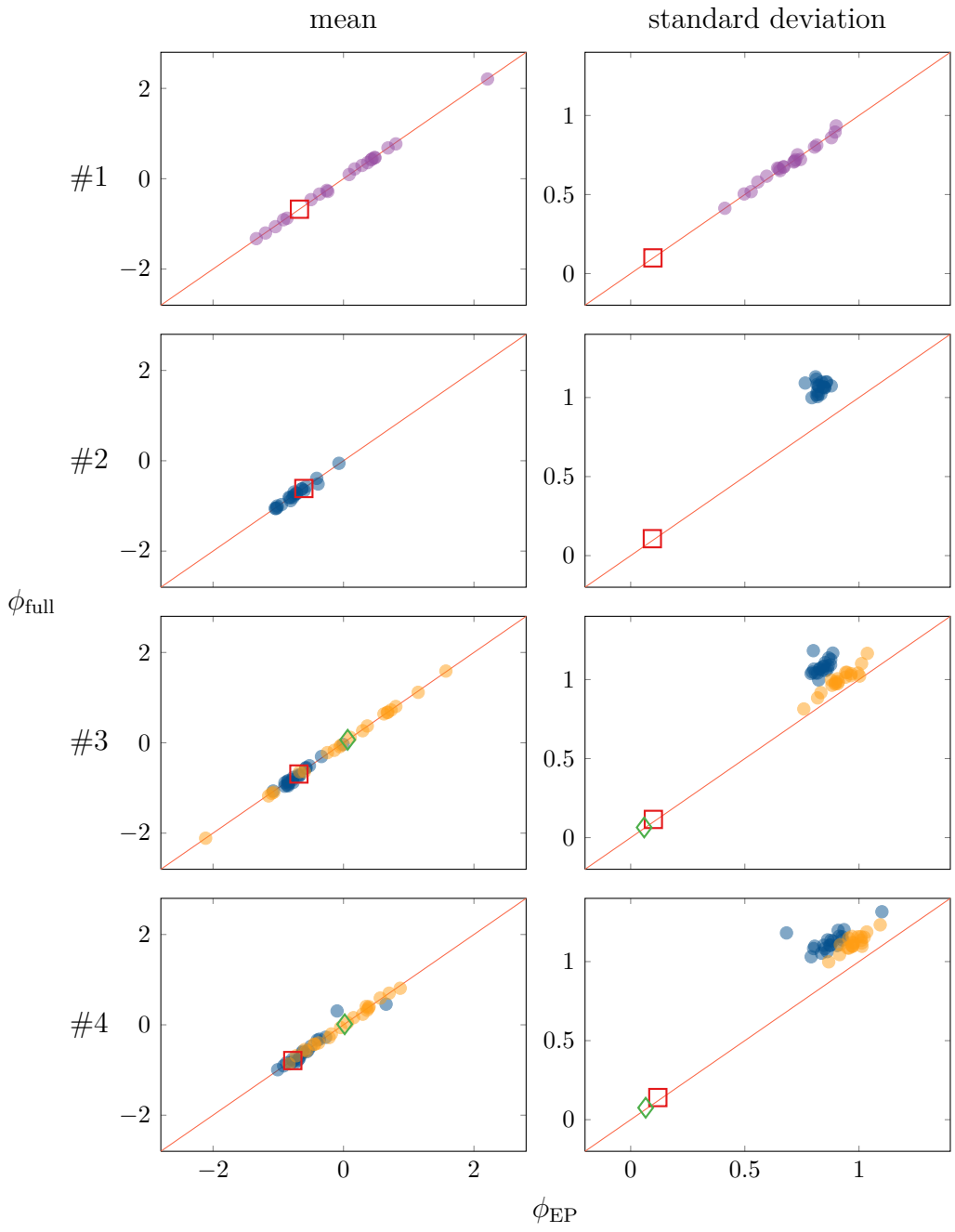


Figure 6.7: Comparison of the posterior mean and standard deviation of the shared parameters between the EP (ϕ_{EP}) and the full (ϕ_{full}) approximation in the classification problem. In this example, the number of sites $K = 16$, the explanatory variable is correlated and sample estimate is used to approximate tilted distribution moments. The resulting EP approximation has been constructed by mixing the samples from all the sites. Markers and the corresponding parameters are: ◇ μ_α, □ $\log(\sigma_\alpha)$, ● $\boldsymbol{\mu}_\beta$, ● $\log(\boldsymbol{\sigma}_\beta)$ and ● $\boldsymbol{\beta}$. The red diagonal line shows the points of equivalence. Rows corresponds to different model structures.

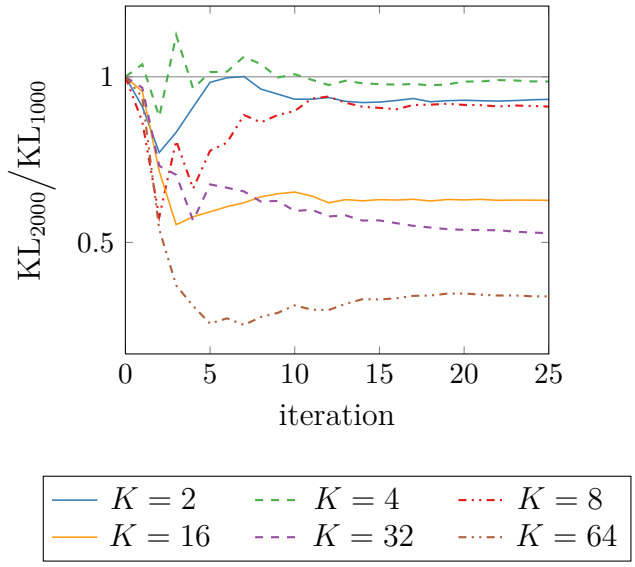


Figure 6.8: The ratio of the KL-divergence with 2000 and 1000 samples per site inference for each iteration. The problem is the linear regression problem with model structure #3. Sample estimate is used to approximate tilted distribution moments.

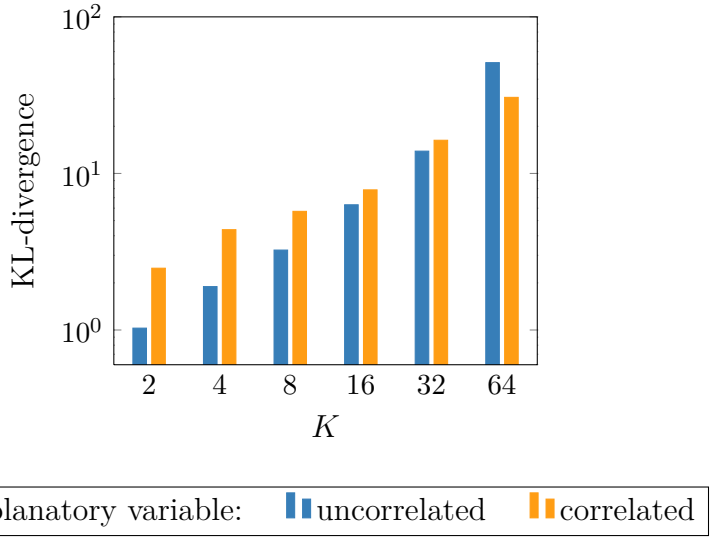


Figure 6.9: Comparison of $\mathrm{KL}(\mathrm{N}_{\mathrm{full}} \| \mathrm{N}_{\mathrm{EP}})$ between the results obtained for uncorrelated and correlated explanatory variable for different number of groups. The problem is the linear regression problem with model structure #3. Sample estimate is used to approximate tilted distribution moments. The y-axis is in the logarithmic scale.

Chapter 7

Discussion

Using EP for distributed Bayesian inference offers many benefits but also introduces possible issues. As the site terms become more complex, it is not generally possible to obtain unbiased estimates for the precision matrix of the tilted distribution. The noise and the bias in these estimates might affect the outcome of the algorithm. The applicability of the method into specific problems must be carefully inspected. However, as shown in the experiments in the chapter 6 and in the results in the section 6.5, it is possible to obtain good results with biased estimates.

The effect of damping factor is not tested in these experiments. It is clear, that the decay rate of the damping factor affect the speed of convergence. However, the effect of damping rate to the resulting approximation requires further analysis. It is likely that, if the tilted distribution moment estimates are relatively noisy and biased, major changes in the decay rate of the damping factor may affect the error in the resulting approximation.

It was illustrated in the figure 6.8 that the increase in the moment estimates increase the approximation error. Here it must be noted that the settings for the damping factor were the same between the experiments. By adjusting the damping factor, it might be possible get results with equal approximation error but with different convergence speed.

Different variance reduction methods can be used to decrease the noise in the estimates of the tilted distribution moments and thus improve the results. However, as these methods may also increase the bias in the estimates, further analysis on their applicability must be conducted. For sample based methods, more straightforward way of reducing the variance is to increase the number of samples. However, unlike some of the other variance reduction methods, this increase in the number of samples might also increase the computational complexity considerably.

One of the main benefits of the method is the scalability. However, as seen from the figure 6.3, increasing the number of sites increases also the error. On

the other hand, as seen from the figure 6.2, increasing the number of sites makes it possible to use bigger step size and thus the individual site inferences might converge faster. The behaviour of the error or the step size with large number of sites is not analysed in this thesis. Thus, further analysis would be required to make conclusions about the scalability.

Chapter 8

Conclusions

EP is usually applied to Bayesian inference problems by factoring the data points into the sites pointwise. Partitioning the points into the sites in bigger sets instead offers more possibilities for distributed inference. Choosing the optimal factorisation is a complex problem. Bigger sites yields more information on each inference so that the number of required iterations is likely to decrease. However, smaller sites are faster to make inference upon. In addition, using fewer sites introduce less approximation error.

In addition to the EP, various methods exists for distributed Bayesian inference. However, they all share the same downside: the inference in the separate partitions is conducted independent on the others. EP performs the inference iteratively and uses the message passing feature to include information from the other partitions or sites into the inference of the others in the next iteration via the cavity distribution. The same approach can be incorporated into the other distributed EP methods by iterating the method and using the posterior distribution as the prior for the next iteration.

EP algorithm considers the likelihood of each site with the approximation of the other sites and updates the corresponding site approximation by moment matching. This requires that the tilted distribution moments can be approximated. Sampling with MCMC methods provides a versatile method for performing this approximation. However, it is not generally possible to obtain an unbiased estimate of the precision matrix from samples. This may introduce error in the resulting approximation or it may prevent the algorithm from converging. However, as shown by the results of the experiments in this thesis, good results can be obtained by using biased moment estimates.

Various variance reduction methods can be applied to the tilted distribution moment estimation. By reducing the noise in the estimates, such methods may help to reduce the error in the resulting EP approximation or speed up the convergence. However, they may also introduce bias and care should be taken when using them.

The advantage of using EP for distributed Bayesian inference, when compared to the other methods with independent site inferences, lies in the message passing feature. Because of it, the computational effort in MCMC methods can be concentrated on the important areas. Areas contradicted by other sites get only small weight in the cavity distribution and are thus payed less attention to in the site inference.

When EP is applied for hierarchical models, the problem may be simplified. The information of the local parameters of one hierarchical group does not affect the inference of the other groups. If one group is inferred only in one site, the information of the corresponding local parameters do not need to be passed for the other sites. Thus the local parameters can be omitted from the parameters included in the EP message passing feature. This decreases the computational complexity of the algorithm greatly.

Appendix A

Sample covariance matrix rank

Consider n samples $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n$ from d-dimensional random variable $\boldsymbol{\theta}$ and the corresponding $d \times n$ sample data matrix $\boldsymbol{\Theta}$ obtained by concatenating all the samples column-wise. The $d \times d$ sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$ is defined by

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n-1} \sum_{i=1}^{n} \Big(\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}\Big)\Big(\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}\Big)^{\mathrm{T}} = \frac{1}{n-1} \Big(\boldsymbol{\Theta} - \bar{\boldsymbol{\theta}} \mathbb{1}_n^{\mathrm{T}}\Big)\Big(\boldsymbol{\Theta} - \bar{\boldsymbol{\theta}} \mathbb{1}_n^{\mathrm{T}}\Big)^{\mathrm{T}}, \tag{A.1}$$

where $\bar{\boldsymbol{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\theta}_i$ is the sample mean. This appendix presents an upper bound for the rank of $\widehat{\boldsymbol{\Sigma}}$.

Lemma A.1. $\operatorname{rank}(\boldsymbol{A}) = \operatorname{rank}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$, $\forall \boldsymbol{A} \in \mathbb{R}^{m,n}$.

Proof. Let $\boldsymbol{x} \in \mathbb{R}^m$. It is clear that $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 0 \Rightarrow \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 0$. If we assume that $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 0$, then $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 0$ and $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}) = 0$, and further $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 0$. Thus $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 0 \Leftrightarrow \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 0, \forall \boldsymbol{x}$, that is $\operatorname{nul}(\boldsymbol{A}^{\mathrm{T}}) = \operatorname{nul}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$. From the Rank–nullity theorem, we get

$$\operatorname{nul}\Big(\boldsymbol{A}^{\mathrm{T}}\Big) + \operatorname{rank}\Big(\boldsymbol{A}^{\mathrm{T}}\Big) = m = \operatorname{nul}\Big(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\Big) + \operatorname{rank}\Big(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\Big).$$

Because the rank of a matrix is the same as the rank of its transpose, eliminating $\operatorname{nul}(\boldsymbol{A}^{\mathrm{T}})$ and $\operatorname{nul}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$ gives $\operatorname{rank}(\boldsymbol{A}) = \operatorname{rank}(\boldsymbol{A}^{\mathrm{T}}) = \operatorname{rank}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$. □

Theorem A.2. $\operatorname{rank}\Big(\widehat{\boldsymbol{\Sigma}}\Big) \leq \min(d, n)$.

Proof. From the lemma A.1, we get $\operatorname{rank}\Big(\widehat{\boldsymbol{\Sigma}}\Big) = \operatorname{rank}\Big(\boldsymbol{\Theta} - \bar{\boldsymbol{\theta}}\mathbb{1}_n^{\mathrm{T}}\Big)$. Because the rank of a matrix can not be greater than its smaller dimension, $\operatorname{rank}\Big(\widehat{\boldsymbol{\Sigma}}\Big) \leq \min(d, n)$. □

Appendix B

Control variates

This appendix presents the control variates method and applies it to the estimation of moments from samples. Consider a multidimensional random variable of which we have independent samples Z. Let random variable $\boldsymbol{f}(Z)$ be an unbiased estimate vector for some unknown quantity of interest $\mathrm{E}(\boldsymbol{f}(Z)) = \boldsymbol{q_f}$ and let $\boldsymbol{h}(Z)$ be a vector of random variables with known expectation $\mathrm{E}(\boldsymbol{h}(Z)) = \boldsymbol{q_h}$. Using these control variates, another unbiased estimator for $\boldsymbol{q_f}$ can be constructed by

$$\widehat{\boldsymbol{f}}(Z) = \boldsymbol{f}(Z) - \boldsymbol{C}^{\mathrm{T}}\big(\boldsymbol{h}(Z) - \boldsymbol{q_h}\big) \tag{B.1}$$

with any choice of conformable matrix $\boldsymbol{C}$. The optimal coefficient $\boldsymbol{C}$, that minimises the variance of $\widehat{\boldsymbol{f}}(Z)$, is

$$\boldsymbol{C}^* = \Big[\mathrm{Var}\big(\boldsymbol{h}(Z)\big)\Big]^{-1} \mathrm{Cov}\big(\boldsymbol{h}(Z), \boldsymbol{f}(Z)\big) \tag{B.2}$$

In practice, the optimal coefficient $\boldsymbol{C}^*$ is not known in advance, but has to be estimated from the samples. If $\boldsymbol{h}(Z)$ and $\boldsymbol{f}(Z)$ are correlated, the variance of $\widehat{\boldsymbol{f}}(Z)$ can be smaller than of $\boldsymbol{f}(Z)$. The more correlated the variables are, the better variance reduction is achieved. Some results on the reduced variance are given for example by Lavenberg, Moeller and Welch (1982).

Consider the case of estimating the first and the second moment of the distribution of $\boldsymbol{\theta}$ from samples $\boldsymbol{\Theta}$. Assume that a distribution $q(\boldsymbol{\theta})$ with known mean $\boldsymbol{\mu}_q$ and covariance $\boldsymbol{\Sigma}_q$ is available and assume that it correlates with the true distribution $\mathrm{p}(\boldsymbol{\theta})$. Using this distribution, control variates can be constructed by weighting the corresponding samples with the ratio of the probabilities of the two distributions, that is using importance sampling to estimate the known moments. The first moment can be estimated by setting

$$\boldsymbol{f}(\boldsymbol{\Theta}) = \bar{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\theta}_i, \tag{B.3}$$

$$\boldsymbol{h}(\boldsymbol{\Theta}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\theta}_i \frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)}. \tag{B.4}$$

The second moment can be estimated by setting

$$\boldsymbol{f}'(\boldsymbol{\Theta}) = \widehat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{\theta}_i - \widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}})(\boldsymbol{\theta}_i - \widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}})^{\mathrm{T}}, \tag{B.5}$$

$$\boldsymbol{h}'(\boldsymbol{\Theta}) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\theta}_i - \boldsymbol{\mu}_q)(\boldsymbol{\theta}_i - \boldsymbol{\mu}_q)^{\mathrm{T}}\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)}, \tag{B.6}$$

and by reshaping $\boldsymbol{f}'(\boldsymbol{\Theta})$ and $\boldsymbol{h}'(\boldsymbol{\Theta})$ into some vectors $\boldsymbol{f}(\boldsymbol{\Theta})$ and $\boldsymbol{h}(\boldsymbol{\Theta})$ for example by flattening the upper triangulars of the matrices. The mean estimate $\widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ can be either the sample mean $\bar{\boldsymbol{\theta}}$ or the improved control variates estimate $\widehat{\boldsymbol{f}}(\boldsymbol{\Theta})$ for the first moment calculated before. The unbiased estimator (B.5) is divided by $n-1$ because $\widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ is also estimated from the samples whereas the estimator (B.6) is divided by n because $\boldsymbol{\mu}_q$ is known.

If the correlation between $\boldsymbol{f}(\boldsymbol{\Theta})$ and $\boldsymbol{h}(\boldsymbol{\Theta})$ is low, using control variates is not beneficial. Thus, additional regularisation can be applied to determine if normal sample estimates or other estimates should be used instead. This can be done for example by monitoring the probabilities $q(\boldsymbol{\theta}_i)$ and/or the coefficient $\boldsymbol{C}$ in various ways.

The estimation of the optimal coefficient $\boldsymbol{C}^*$ is presented in the following two subsections for the first and the second moment. These estimates may be quite imprecise but that does not impact too much on the accuracy of the moment estimates.

First moment

For the first moment, the estimates $\boldsymbol{f}(\boldsymbol{\Theta})$ and $\boldsymbol{h}(\boldsymbol{\Theta})$ are:

$$\boldsymbol{f}(\boldsymbol{\Theta}) = \bar{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\theta}_i, \qquad \boldsymbol{h}(\boldsymbol{\Theta}) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\theta}_i\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)}.$$

The required terms for the optimal coefficient can be estimated by

$$\begin{aligned}
\operatorname{Var}\big(\boldsymbol{h}(\boldsymbol{\Theta})\big) &= \frac{1}{n^2}\sum_{i=1}^{n}\operatorname{Var}\left(\boldsymbol{\theta}_i\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)}\right) \\
&= \frac{1}{n}\operatorname{Var}\left(\boldsymbol{\theta}\frac{q(\boldsymbol{\theta})}{\mathrm{p}(\boldsymbol{\theta})}\right) \\
&\approx \frac{1}{n^2}\sum_{i=1}^{n}\left(\boldsymbol{\theta}_i\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)} - \boldsymbol{\mu}_q\right)\left(\boldsymbol{\theta}_i\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)} - \boldsymbol{\mu}_q\right)^{\mathrm{T}},
\end{aligned}$$

$$\begin{aligned}\operatorname{Cov}\big(\boldsymbol{h}(\boldsymbol{\Theta}), \boldsymbol{f}(\boldsymbol{\Theta})\big) &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\operatorname{Cov}\left(\boldsymbol{\theta}_i\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)}, \boldsymbol{\theta}_j\right)\\ &= \frac{1}{n}\operatorname{Cov}\left(\boldsymbol{\theta}\frac{q(\boldsymbol{\theta})}{\mathrm{p}(\boldsymbol{\theta})}, \boldsymbol{\theta}\right)\\ &\approx \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(\boldsymbol{\theta}_i\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)} - \boldsymbol{\mu}_q\right)\big(\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}\big)^{\mathrm{T}}.\end{aligned}$$

When calculating the optimal coefficient $\boldsymbol{C}^*$ given in equation (B.2), the factors $1/n$ cancel out.

Second moment

For the second moment, the estimates $\boldsymbol{f}(\boldsymbol{\Theta})$ and $\boldsymbol{h}(\boldsymbol{\Theta})$ are some reshaped vectorisations from

$$\boldsymbol{f}'(\boldsymbol{\Theta}) = \widehat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{\theta}_i - \widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}})(\boldsymbol{\theta}_i - \widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}})^{\mathrm{T}},$$

and

$$\boldsymbol{h}'(\boldsymbol{\Theta}) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\theta}_i - \boldsymbol{\mu}_q)(\boldsymbol{\theta}_i - \boldsymbol{\mu}_q)^{\mathrm{T}}\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)},$$

respectively. The mean estimate $\widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ can be either the sample mean $\bar{\boldsymbol{\theta}}$ or the improved control variates estimate $\widehat{\boldsymbol{f}}(\boldsymbol{\Theta})$ for the first moment calculated before.

Let mapping $\operatorname{resh}(\cdot)$ denote the selected reshaping from matrix to vector and let $\operatorname{op}(\cdot)$ denote the outer product of a vector with itself, that is $\operatorname{op}(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$. Similarly as in the case of the first moment, the variance of $\boldsymbol{h}(\boldsymbol{\Theta})$ can be estimated directly by

$$\begin{aligned}\operatorname{Var}\big(\boldsymbol{h}(\boldsymbol{\Theta})\big) &= \frac{1}{n}\operatorname{Var}\left[\operatorname{resh}\left(\operatorname{op}(\boldsymbol{\theta} - \boldsymbol{\mu}_q)\frac{q(\boldsymbol{\theta})}{\mathrm{p}(\boldsymbol{\theta})}\right)\right]\\ &\approx \frac{1}{n^2}\sum_{i=1}^{n}\operatorname{op}\left[\operatorname{resh}\left(\operatorname{op}(\boldsymbol{\theta}_i - \boldsymbol{\mu}_q)\frac{q(\boldsymbol{\theta}_i)}{\mathrm{p}(\boldsymbol{\theta}_i)} - \boldsymbol{\Sigma}_q\right)\right].\end{aligned}$$

Because the sample mean is included in the estimate $\boldsymbol{f}(\boldsymbol{\Theta})$, the covariance between $\boldsymbol{h}(\boldsymbol{\Theta})$ and $\boldsymbol{f}(\boldsymbol{\Theta})$ can not be factored into independent components. Thus an additional partitioning of the samples into n_p disjoint sets $P_1, P_2, \ldots, P_{n_p}$ is required. For simplicity, let us assume that each set has equally many samples n_s. The covariance $\boldsymbol{S}_{n_s} = \operatorname{Cov}\big(\boldsymbol{h}_{n_s}(\boldsymbol{\Theta}), \boldsymbol{f}_{n_s}(\boldsymbol{\Theta})\big)$ for estimates formed with n_s samples can be estimated by simply evaluating $\boldsymbol{h}(\boldsymbol{\Theta})$ and $\boldsymbol{f}(\boldsymbol{\Theta})$ for each set P_i and calculating the respective sample variance $\widehat{\boldsymbol{S}}_{n_s}$. As the variance of sample variance in general is

$$\operatorname{Var}\big(s^2\big) = \frac{1}{n}\left(\mu_4 - \frac{n-3}{n-1}\sigma_4\right),$$

where μ_4 is the fourth moment about the mean, the estimate $\widehat{\boldsymbol{S}}_{n_s}$ has to be scaled to the total number of samples n by

$$\mathrm{Cov}\big(\boldsymbol{h}(\boldsymbol{\Theta}), \boldsymbol{f}(\boldsymbol{\Theta})\big) = \frac{n_s}{n}\widehat{\boldsymbol{S}}_{n_s} + \left(\frac{n_s - 3}{n_s - 1} - \frac{n - 3}{n - 1}\right)\frac{1}{n}\widehat{\boldsymbol{\Sigma}}.$$

Appendix C

Coefficient of determination

Consider linear regression model defined in equation (6.1). The response variable can be denoted by

$$y = F + \varepsilon, \tag{C.1}$$

where $F = f(\boldsymbol{x}) = \alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x},\ \varepsilon \sim \mathrm{N}(0, \sigma)$ and

$$\mathrm{Cov}(F, \varepsilon) = 0. \tag{C.2}$$

The coefficient of determination R^2 is defined as

$$R^2 = [\mathrm{Cor}(y, F)]^2 = \frac{[\mathrm{Cov}(y, F)]^2}{\mathrm{Var}(y)\,\mathrm{Var}(F)}. \tag{C.3}$$

The covariance between y and F can be expanded to

$$\begin{aligned}
\mathrm{Cov}(y, F) &= \mathrm{E}(yF) - \mathrm{E}(y)\,\mathrm{E}(F) \\
&= \mathrm{E}\Big(F^2 + F\varepsilon\Big) - \Big(\mathrm{E}(F) + \mathrm{E}(\varepsilon)\Big)\,\mathrm{E}(F) \\
&\overset{\text{(C.2)}}{=} \mathrm{E}\Big(F^2\Big) + \mathrm{E}(F)\,\mathrm{E}(\varepsilon) - [\mathrm{E}(F)]^2 - \mathrm{E}(F)\,\mathrm{E}(\varepsilon) \\
&= \mathrm{E}\Big(F^2\Big) - [\mathrm{E}(F)]^2 \\
&= \mathrm{Var}(F),
\end{aligned}$$

Notice that the above equation holds even if the noise term ε would not have zero mean. By substituting this into equation (C.3), R^2 can be simplified to

$$R^2 = \frac{[\mathrm{Var}(F)]^2}{\mathrm{Var}(y)\,\mathrm{Var}(F)} = \frac{\mathrm{Var}(F)}{\mathrm{Var}(y)}. \tag{C.4}$$

Bibliography

Ahn, S., Korattikara, A. and Welling, M. (2012). "Bayesian posterior sampling via stochastic gradient Fisher scoring". In: *Proceedings of the 29th International Conference on Machine Learning*. Omnipress, pp. 1591–1598.

Al-Baali, M., Spedicato, E. and Maggioni, F. (2014). "Broyden's quasi-Newton methods for a nonlinear system of equations and unconstrained optimization: A review and open problems". In: *Optimization Methods and Software* 29.5, pp. 937–954.

Bai, J. and Shi, S. (2011). "Estimating high dimensional covariance matrices and its applications". In: *Annals of Economics and Finance* 12.2, pp. 199–215.

Barthelmé, S. and Chopin, N. (2014). "Expectation propagation for likelihood-free inference". In: *Journal of the American Statistical Association* 109.505, pp. 315–333.

Beal, M. J. (2003). "Variational algorithms for approximate Bayesian inference". PhD thesis. University College London.

Betancourt, M. J. (2013). "A general metric for Riemannian manifold Hamiltonian Monte Carlo". In: *Geometric Science of Information: First International Conference, GSI 2013, Paris, France, August 28-30, 2013, Proceedings*. Vol. 8085. Lecture Notes in Computer Science. Springer, pp. 327–334.

Betancourt, M. J. (2014). *Adiabatic Monte Carlo*. arXiv:`1405.3489 [stat.ME]`.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Information Science and Statistics. Springer.

Bodnar, T. and Gupta, A. K. (2011). "Estimation of the precision matrix of a multivariate elliptically contoured stable distribution". In: *Statistics* 45.2, pp. 131–142.

Bodnar, T., Gupta, A. K. and Parolya, N. (2014a). "On the strong convergence of the optimal linear shrinkage estimator for large dimensional covariance matrix". In: *Journal of Multivariate Analysis* 132, pp. 215–228.

Bodnar, T., Gupta, A. K. and Parolya, N. (2014b). *Optimal linear shrinkage estimator for large dimensional precision matrix*. arXiv:`1308.0931 [math.ST]`.

Boyle, P. P. (1977). "Options: A Monte Carlo approach". In: *Journal of Financial Economics* 4.3, pp. 323–338.

Cichocki, A. and Amari, S. (2010). "Families of alpha- beta- and gamma-divergences: Flexible and robust measures of similarities". In: *Entropy* 12.6, pp. 1532–1568.

Cornuet, J., Marin, J.-M., Mira, A. and Robert, C. P. (2012). "Adaptive multiple importance sampling". In: *Scandinavian Journal of Statistics* 39.4, pp. 798–812.

Cseke, B. and Heskes, T. (2011). "Approximate marginals in latent Gaussian models". In: *The Journal of Machine Learning Research* 12, pp. 417–454.

Duane, S., Kennedy, A. D., Pendleton, B. J. and Roweth, D. (1987). "Hybrid Monte Carlo". In: *Physics letters B* 195.2, pp. 216–222.

Friedman, J., Hastie, T. and Tibshirani, R. (2008). "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3, pp. 432–441.

Gelman, A., Vehtari, A., Jylänki, P., Robert, C., Chopin, N. and Cunningham, J. P. (2014a). *Expectation propagation as a way of life*. arXiv:1412.4869 [stat.CO].

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2014b). *Bayesian data analysis*. Taylor & Francis Group, CRC Press.

Gershman, S., Hoffman, M. and Blei, D. (2012). "Nonparametric variational inference". In: *Proceedings of the 29th International Conference on Machine Learning*. Omnipress, pp. 663–670.

Geweke, J. (1989). "Bayesian inference in econometric models using Monte Carlo integration". In: *Econometrica* 57.6, pp. 1317–1339.

Gupta, A. K., Varga, T. and Bodnar, T. (2013). *Elliptically contoured models in statistics and portfolio theory*. Springer.

Hastings, W. K. (1970). "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1, pp. 97–109.

Hinde, J. (2014). "Logistic Normal Distribution". In: *International Encyclopedia of Statistical Science*. Ed. by M. Lovric. Springer Berlin, pp. 754–755.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013). "Stochastic variational inference". In: *Journal of Machine Learning Research* 14, pp. 1303–1347.

Homan, M. D. and Gelman, A. (2014). "The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo". In: *The Journal of Machine Learning Research* 15.1, pp. 1593–1623.

Jylänki, P., Vanhatalo, J. and Vehtari, A. (2011). "Robust Gaussian process regression with a student-t likelihood". In: *Journal of Machine Learning Research* 12, pp. 3227–3257.

Korattikara, A., Chen, Y. and Welling, M. (2014). "Austerity in MCMC land: Cutting the Metropolis-Hastings budget". In: *Proceedings of the 31st International Conference on Machine Learning*. Omnipress, pp. 321–336.

Kullback, S (1959). *Information Theory and Statistics*. Dover Publications.

Kullback, S. and Leibler, R. A. (1951). "On information and sufficiency". In: *Annals of Mathematical Statistics* 22.1, pp. 79–86.

Kuss, M. and Rasmussen, C. E. (2005). "Assessing approximations for Gaussian process classification". In: *Advances in Neural Information Processing Systems 18*. MIT Press, pp. 699–706.

Lavenberg, S. S., Moeller, T. L. and Welch, P. D. (1982). "Statistical results on control variables with application to queueing network simulation". In: *Operations Research* 30.1, pp. 182–202.

Le Cam, L. and Yang, G. L. (2000). *Asymptotics in statistics: Some basic concepts*. Springer.

Ledoit, O. and Wolf, M. (2004). "A well-conditioned estimator for large-dimensional covariance matrices". In: *Journal of Multivariate Analysis* 88.2, pp. 365–411.

Ledoit, O. and Wolf, M. (2012). "Nonlinear shrinkage estimation of large-dimensional covariance matrices". In: *Annals of Statistics* 40.2, pp. 1024–1060.

Lewandowski, D., Kurowicka, D. and Joe, H. (2009). "Generating random correlation matrices based on vines and extended onion method". In: *Journal of Multivariate Analysis* 100.9, pp. 1989–2001.

Metropolis, N. and Ulam, S. (1949). "The Monte Carlo method". In: *Journal of the American Statistical Association* 44.247, pp. 335–341.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). "Equation of state calculations by fast computing machines". In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092.

Minka, T. P. (2001a). "A Family of algorithms for approximate Bayesian inference". PhD thesis. Massachusetts Institute of Technology.

Minka, T. P. (2001b). "Expectation propagation for approximate Bayesian inference". In: *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., pp. 362–369.

Minka, T. P. (2004). *Power EP*. Tech. rep. Microsoft Research, Cambridge.

Muirhead, R. J. (2005). *Aspects of multivariate statistical theory*. John Wiley & Sons.

Neal, R. M. (1994). "An improved acceptance procedure for the hybrid Monte Carlo algorithm". In: *Journal of Computational Physics* 111.1, pp. 194–203.

Neiswanger, W., Wang, C. and Xing, E. (2014). *Asymptotically exact, embarrassingly parallel MCMC*. arXiv:`1311.4780 [stat.ML]`.

Opper, M. and Winther, O. (2000). "Gaussian processes for classification: Mean-field algorithms". In: *Neural Computation* 12.11, pp. 2655–2684.

Parlett, B. N. and Scott, D. S. (1979). "The Lanczos algorithm with selective orthogonalization". In: *Mathematics of computation* 33.145, pp. 217–238.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Pas-

sos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). "Scikit-learn: Machine learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Riihimäki, J., Jylänki, P. and Vehtari, A. (2013). "Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood". In: *Journal of Machine Learning Research* 14, pp. 75–109.

Robbins, H. and Monro, S. (Sept. 1951). "A stochastic approximation method". In: *The Annals of Mathematical Statistics* 22.3, pp. 400–407.

Rousseeuw, P. J. (1984). "Least median of squares regression". In: *Journal of the American Statistical Association* 79.388, pp. 871–880.

Rousseeuw, P. J. and Driessen, K. V. (1999). "A fast algorithm for the minimum covariance determinant estimator". In: *Technometrics* 41.3, pp. 212–223.

Sarr, A. and Gupta, A. K. (2009). "Estimation of the precision matrix of multivariate Kotz type model". In: *Journal of Multivariate Analysis* 100.4, pp. 742–752.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H, George, E and McCulloch, R (2013). "Bayes and big data: The consensus Monte Carlo algorithm". In: *Bayes 250*. International Society for Bayesian Analysis.

Sivula, T. (2014). *EP-Stan*. URL: `https://github.com/gelman/ep-stan` (visited on 18th Mar. 2015).

Smola, A. J., Vishwanathan, S. V. N. and Eskin, E. (2004). "Laplace propagation". In: *Advances in Neural Information Processing Systems 16*. MIT Press, pp. 441–448.

Stan Development Team (2014a). *PyStan: the Python interface to Stan, Version 2.5.0*. URL: `http://mc-stan.org/pystan.html` (visited on 18th Mar. 2015).

Stan Development Team (2014b). *Stan: A C++ Library for Probability and Sampling, Version 2.5.0*. URL: `http://mc-stan.org/` (visited on 18th Mar. 2015).

Stein, C. (1956). "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution". In: *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1: Contributions to the Theory of Statistics. University of California Press, pp. 197–206.

Strecok, A. (1968). "On the calculation of the inverse of the error function". In: *Mathematics of Computation* 22.101, pp. 144–158.

Tierney, L., Kass, R. E. and Kadane, J. B. (1989). "Fully exponential Laplace approximations to expectations and variances of nonpositive functions". In: *Journal of the American Statistical Association* 84.407, pp. 710–716.

Tierney, L. and Kadane, J. B. (1986). "Accurate approximations for posterior moments and marginal densities". In: *Journal of the American Statistical Association* 81.393, pp. 82–86.

Tsukuma, H. and Konno, Y. (2006). "On improved estimation of normal precision matrix and discriminant coefficients". In: *Journal of multivariate Analysis* 97.7, pp. 1477–1500.

Villani, M. and Larsson, R. (2006). "The multivariate split normal distribution and asymmetric principal components analysis". In: *Communications in Statistics - Theory and Methods* 35.6, pp. 1123–1140.

Wang, C. and Blei, D. M. (2013). "Variational inference in nonconjugate models". In: *Journal of Machine Learning Research* 14, pp. 899–925.

Wang, X. and Dunson, D. B. (2014). *Parallelizing MCMC via Weierstrass sampler*. arXiv:1312.4605 [stat.CO].

Witten, D. M., Friedman, J. H. and Simon, N. (2011). "New insights and faster computations for the graphical lasso". In: *Journal of Computational and Graphical Statistics* 20.4, pp. 892–900.

Zhang, T. and Zou, H. (2014). "Sparse precision matrix estimation via lasso penalized D-trace loss". In: *Biometrika* 101.1, pp. 103–120.