Journal of Machine Learning Research 9 (2008) 759-813 Submitted 10/07; Revised 2/08; Published 4/08

Bayesian Inference and Optimal Design for the Sparse Linear Model

Matthias W. Seeger SEEGER@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics
Spemannstr. 38, Tübingen, Germany

Editor: Martin Wainwright

Abstract

The linear model with sparsity-favouring prior on the coefficients has important applications in many different domains. In machine learning, most methods to date search for maximum a posteriori sparse solutions and neglect to represent posterior uncertainties. In this paper, we address problems of Bayesian optimal design (or experiment planning), for which accurate estimates of uncertainty are essential. To this end, we employ expectation propagation approximate inference for the linear model with Laplace prior, giving new insight into numerical stability properties and proposing a robust algorithm. We also show how to estimate model hyperparameters by empirical Bayesian maximisation of the marginal likelihood, and propose ideas in order to scale up the method to very large underdetermined problems.

We demonstrate the versatility of our framework on the application of gene regulatory network identification from micro-array expression data, where both the Laplace prior and the active experimental design approach are shown to result in significant improvements. We also address the problem of sparse coding of natural images, and show how our framework can be used for compressive sensing tasks.

Part of this work appeared in Seeger et al. (2007b). The gene network identification application appears in Steinke et al. (2007).

Keywords: sparse linear model, Laplace prior, expectation propagation, approximate inference, optimal design, Bayesian statistics, gene network recovery, image coding, compressive sensing

1. Introduction

In many settings favoured in current machine learning work, the model and data set are given in advance, and predictions with low error are sought. Many methods from different paradigms have successfully been applied to these problems. While Bayesian approaches, such as the one we describe here, enjoy some benefits in this regime, they can be more difficult to implement, less algorithmically robust, and often require more computation time than, for example, penalised estimation methods, whose computation often reduces to a standard optimisation problem. In our opinion, the real practical power of the Bayesian way is revealed better in higher-level tasks such as making optimally cost-efficient decisions or *experimental design*. In the latter, aspects of the model and measurement experiments are adapted based on growing knowledge about the current situation, and data is sampled in a sequential and actively controlled manner, with the aim of obtaining answers as quickly as possible. Our main motivation in the present work is to demonstrate how Bayesian experimental design can be implemented in a computationally efficient and robust way, and how a range of challenging applications can benefit from selectively sampling data where it is most needed.

A number of characteristics of the framework we propose here, are especially useful, if not essential, to drive efficient experimental design for the applications we consider. The latter, at least in the sequential variant discussed here, proceeds through a significant number of individual decisions (say, where to sample data next). In order to make each decision, our current uncertainty in variables of interest needs to be estimated quantitatively, and for a large number of candidates we have to consider how, and by how much, each of them would reduce this uncertainty estimate. As will become clear in the sequel, the uncertainty estimate is given by the posterior distribution, an approximation to which can be obtained robustly and efficiently by our method. The estimate is given as a Gaussian distribution, whose change after one more experiment can robustly and very efficiently be quantified. These points motivate our insistence on robustness[1] and efficiency below. Another key aspect of the models treated here is sparsity. This regularisation principle allows us to start from an overparameterised model, forcing parameters close to zero if they are not required. In our experiments, we demonstrate that the interplay between sparsity regularisation and experimental design seems to be particularly successful. In sequential design, most of the decisions have to be done early, without a lot of data available, and the focus (under a sparsity prior) on a few relevant effects only seems particularly useful in that respect.[2] In contrast, if the models of interest here are used with Gaussian priors, as is usually done, then sequential design is not different from optimising X beforehand. Although observations become available along the way, these are not used at all. We come back to this important point below.

In this work, we consider the *linear model*

$$u = Xa + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I), \tag{1}$$

where $X \in \mathbb{R}^{m,n}$ is the design matrix, and $a \in \mathbb{R}^n$ is the vector of unknown parameters (or weights). σ^2 is the variance of the Gaussian noise. The model can be thought of as representing a noisy linear system. It is called *underdetermined* if $m \leq n$, and *overdetermined* otherwise. In the underdetermined case, there are in general many solutions, even if we did not allow for noise, and additional desired qualities of a need to be formalised. In a Bayesian framework, this is done by placing a *prior distribution* on a, concentrating its mass on parameters fulfilling the requirements.

In the applications we consider, sparsity of a is a key prior assumption: elements of a should be set to very small values whenever they are not required to describe the data well. On the other hand, few elements should be allowed to be large if necessary. Among different solutions, the ones with the largest number of very small components should be preferred *a priori*. Enforcing sparsity is a fundamental statistical regularisation principle and lies behind many well known ideas such as *selective shrinkage* or *feature selection*. It is discussed in more detail in Section 2.1. Many sparsity-favouring priors have been suggested in statistics. In this paper, we concentrate on independent *Laplace* (or *double exponential*) distribution priors of the form

$$P(a) = \prod_i P(a_i), \quad P(a_i) = \frac{\tilde{\tau}}{2} e^{-\tilde{\tau}|a_i|}, \quad \tilde{\tau} = \tau/\sigma. \tag{2}$$

1. Robustness is an issue which is often overlooked when comparing machine learning methods, yet it is quite essential in experimental design, where many decisions have to be done based on small posterior changes, and where non-robust methods often lead to undesired, erratic high-variance behaviour. In experimental design, robustness can be more important than high posterior approximation accuracy.
2. We report empirical observations here at the moment. We are not aware of strong theoretical results about this aspect.

A key advantage of this choice over others is log-concavity, which implies important computational advantages (see Section 2.1, Section 3.5). We refer to the linear model with Laplace prior as *sparse linear model*.[3]

It is important to note that our method here is different from most of the classical treatments of experimental design for the linear model, which entirely focus on Gaussian prior distributions. The difference to these approaches lies in our use of non-Gaussian sparsity priors. Bayesian inference for the linear model with Gaussian prior is analytically tractable (see O'Hagan, 1994, Chapter 9), and most of the algorithmic complications we address in the following, do not arise there. On the other hand, comparative results in some of our experiments show very significant benefits of using experimental design with sparsity priors rather than Gaussian ones. Our findings point out the need to theoretically analyse and understand experimental design with non-Gaussian priors, although in the absence of analytically tractable formulae for inference, such studies would have to be done conditioned on particular inference approximations.

Once the linear model is endowed with sparsity priors which are not Gaussian, Bayesian inference in general is not analytically tractable anymore and has to be approximated. In this paper, we employ the expectation propagation (EP) algorithm (Minka, 2001b; Opper and Winther, 2000) for approximate Bayesian inference in the sparse linear model. Our motivation runs contrary to most machine learning applications of the sparse linear model considered so far (where maximally sparse solutions for a given fixed problem are estimated and good uncertainty representations seem unimportant), mainly because Bayesian experimental design is fundamentally driven by such uncertainty representations. While Bayesian inference can also be performed using Markov chain Monte Carlo (MCMC) (Park and Casella, 2005), our approach is much more efficient, especially in the context of sequential design, and can be applied to large-scale problems of interest in machine learning. Moreover, experimental design requires the robust estimation in posterior changes across many candidates, starting from a well-defined current distribution, which seems difficult to do with MCMC. The application of EP to the sparse linear model is numerically challenging, and some novel techniques are introduced here in order to obtain a robust algorithm. In this context, the role of log-concavity for numerical stability of EP is clarified. Moreover, a variant known as fractional EP (or Power EP) (Minka, 2004) is shown to essentially overcome stability problems in the context of underdetermined models, while standard EP seems inherently unworkable in these cases. This observation about fractional EP is novel to our knowledge.

We apply our method to the problem of identifying gene regulatory networks from data obtained through active experiments, disturbing the system in a controlled manner. Since such experiments are expensive and time-consuming, a sequentially designed approach is clearly beneficial. Indeed, our experiments on synthetic data, simulated using realistic setups, show clear advantages in using Bayesian experimental design and sparsity priors over traditional approaches.

We also address the problem of sparse linear coding of natural images, optimising the codebook by empirical Bayesian marginal likelihood maximisation. Since current hypotheses about the development of early visual neurons in the brain are equivalent to a Bayesian sparse linear model setup (Lewicki and Olshausen, 1999), our method is useful to test and further refine these.

There has been a lot of recent interest in signal processing in the problem of compressive sensing (Candès et al., 2006; Donoho, 2006; Ji and Carin, 2007). We show how our framework directly

3. The reader may be puzzled about the parameterisation in terms of $\tilde{\tau} = \tau/\sigma$. One reason for this is that it renders τ scale-free: it does not depend on the scale of the response u. A more important reason is given in Section 3.5.

addresses the key issues there, which are in fact optimal design problems, and we motivate applications.

The structure of this paper is as follows. In Section 2, the statistical notion of sparsity is explained and contrasted with notions currently dominant in machine learning. Furthermore, some key applications of the sparse linear model are described. In Section 3, we show how to do approximate inference using the expectation propagation method. Optimal design is discussed in Section 4, and an approximation to the marginal likelihood is given in Section 5. We show how to address large-scale problems in Section 6. Experimental results are presented in Section 7. Our framework is directly related to other approximate inference techniques in Section 8. The paper closes with a discussion in Section 9.

Efficient and extendible code for the sparse linear model will be put into the public domain, as part of the LHOTSE toolbox for adaptive statistical models.[4]

2. The Role of Sparsity. Applications

In this section, we clarify the statistical role of sparsity and motivate the Laplace prior (2) towards this end. We also introduce the applications of interest in our work here: identification of gene networks, and sparse coding of natural images, and we give remarks about applications to compressive sensing, which are subject to work in progress (Seeger and Nickisch, 2008). The importance of optimal design and hyperparameter estimation are motivated using these examples.

2.1 The Role of Sparsity Priors

In order to obtain flexible inference methods, it often makes sense in statistics to employ models with many more degrees of freedom than could uniquely be adapted given finite data. The resulting under-determinedness (sometimes referred to as "ill-posedness", "curse", or other equally negative terms) is broken by making additional assumptions, leading to the fact that some solutions are preferred over others, although both fit the data equally well. The mechanics of this comes in different variants, such as adding a penalty term (or regulariser) to a data-fit functional, or placing a prior distribution over hypotheses. The underlying principles are, however, the same.

A fundamental regularisation idea is *sparsity*. For example, suppose a prediction function is a linear combination of features. If knowledge of good (or optimal) features for a task is vague, it makes sense to allow for a large number of candidates, then let the data decide which are relevant. A sparsity prior (or regulariser) on the coefficients, for example in the sparse linear model (1) with Laplace prior (2), leads to just that. It is important to contrast this with the different, frequently used idea of forcing components to be uniformly small in size, so that the final predictor is a sum of many (or all) features, with each giving a small but non-zero contribution. An example of the latter is the linear model (1) with a *Gaussian* prior $P(a)$, which due to conjugacy allows for a simple analytical treatment (see O'Hagan, 1994, Chapter 9). Such a prior does not encode sparsity. The Laplace distribution puts much more weight close to zero than the Gaussian, while still having higher probabilities for large values. The implications are depicted in Figure 1, see also Tipping (2001).

A sparsity prior embodies the bi-separation characteristic: such parameters a with many very small components at the expense of few large ones are favoured over a whose components are

4. See www.kyb.tuebingen.mpg.de/bs/people/seeger/lhotse/.

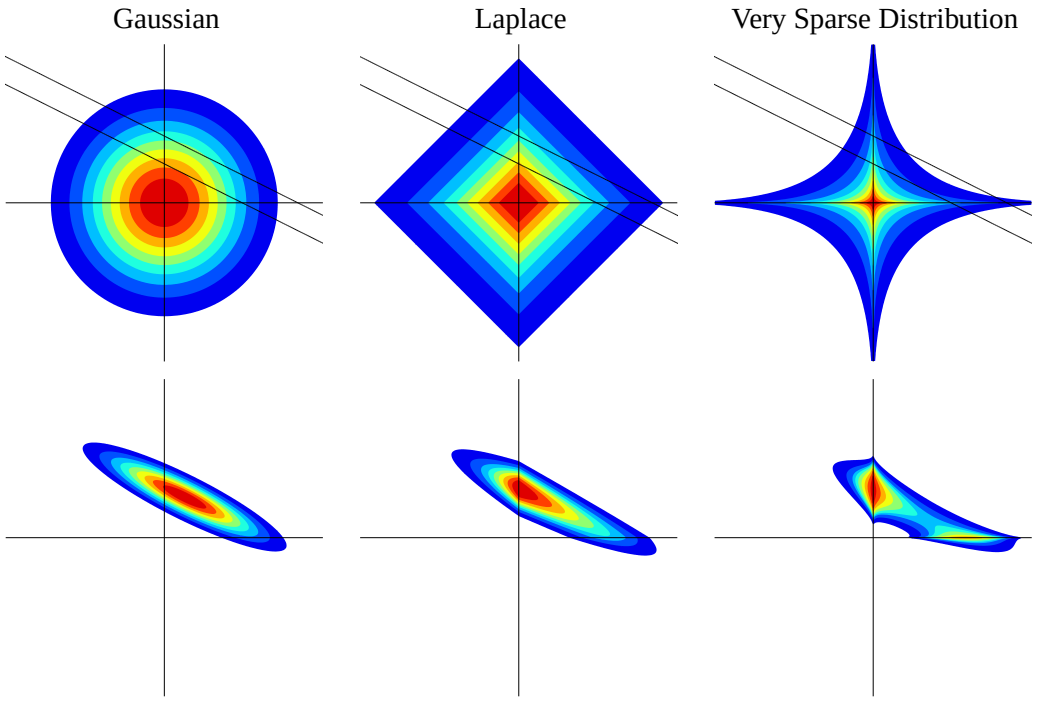

Figure 1: The entries of the parameter a can be given different prior distributions. Shown above are three candidates, plotted jointly over the values of two entries: a Gaussian, a Laplace, and a "very sparse" distribution ($P(a_i) \propto \exp(-\tau|a_i|^{0.4})$). We show contour plots of density functions, where areas of a specific color contain the same probability mass for each of the distributions. The upper row shows prior distributions of unit variance, together with the likelihood for a single measurement (a single linear constraint with Gaussian uncertainty). The lower row shows the corresponding posterior distributions. Whereas the Gaussian prior is spherically distributed, the other two shift probability mass towards the axes, so that more mass is given to sparse tuples (with one entry close to zero). This effect is clearly visible in the posterior distributions, being the normalised product of prior and likelihood. For the Gaussian prior, the areas close to the axes have rather low mass. In comparison, the posterior for the Laplace prior is skewed, so that more mass is concentrated close to the vertical axis. Both posteriors are log-concave and unimodal. The posterior for the "very sparse" prior shows shrinkage towards the axes even more strongly, and in terms of enforcing sparsity, this prior is preferable to the Laplacian. However, the posterior is bimodal now, suggesting two different interpretations for the single observation. The number of posterior modes can increase exponentially with the number of dimensions, so that sampling from or even representing this distribution has combinatorial complexity in general. *Figure by Florian Steinke*.

uniformly small throughout, but sizes are distributed regularly over this "range of smallness". Under the prior, most mass concentrates close to zero, but the tails are also comparatively heavy, allowing for occasional large values. In fact, heavy tails are an essential feature of a sparsity prior, since suppressing many components while still maintaining a flexible range of hypotheses is possible only if some components are allowed to take dominant values. The opposite is true for traditional Gaussian priors. Ishwaran and Rao (2005) call this bi-separation effect *selective shrinkage*, in that

parameters are shrunk towards zero selectively, while a Gaussian prior leads to a more uniform shrinkage. This characteristic is embodied even more strongly in sparsity priors other than the Laplace, such as "spike-and-slab" (mixture of narrow and very wide Gaussian), Student's t, or distributions $\propto \exp(-|\cdot|^\alpha)$, $\alpha < 1$, see also Figure 1. Among these, only the Laplace distribution is *log-concave*, leading to a posterior whose log density is a concave function, thus has a single local maximum. This simplifies and robustifies accurate inference computations significantly (see Section 3.5). For a non-log-concave prior, posteriors tend to be multi-modal, spreading their mass among many bumps, and accurate approximate inference can be a very hard problem. Furthermore, existing variational inference methods are more prone to non-robust unstable behaviour if applied to such models, and convergence or approximation errors can be hard to assess. Since we aim our method to be robust and easy to use by non-experts, we concentrate on log-concave Laplace sparsity priors in the sequel. The importance of log-concavity has been recognised in statistics and Markov chain sampling (Pratt, 1981; Gilks and Wild, 1992; Park and Casella, 2005; Lovász and Vempala, 2003; Paninski, 2005), but has not received much attention so far in work on variational approximate inference.

Our decision to prefer the Laplace sparsity prior over the conventional Gaussian choice, at the expense of having to approximate inference and of introducing significant complications, is ultimately validated by our experimental findings, where the Laplace prior yields large improvements over the Gaussian setting (see Section 7.1). However, apart from failing to encode a sparsity bi-separation, the Gaussian prior leads to other serious artifacts in the context of experimental design with the linear model. For example, suppose we are interested in sequentially designing covariates x for which responses u are queried (this is related to, but not the same setting we use here, see Section 4), say by choosing a "location" t in a feature map $x(t)$. It is well known and easily established that the Bayesian optimal design is *independent* of the response measurements we obtain along the way, it can in fact be computed beforehand. This fact seems absurd for many design problems, including ours here, pointing out a shortcoming of the model-prior combination. In the gene network identification problem (see Section 2.2 for notation), if we were to use a Gaussian prior, the posterior covariances would be identical for all rows of A. This means that no matter what disturbance experiments are done, the uncertainty in how gene i is influenced directly by the others, is the same for all i! Since design decisions mainly hinge on these uncertainty estimates, such artifacts due to a bad prior choice can lead to very suboptimal outcomes (see Section 7.1).

It is important to contrast our approach, and more generally the Bayesian statistical notion of sparsity, with what some *maximum a posteriori* (MAP) treatments of the sparse linear model are aiming to do. In the latter approach, which is very prominent in machine learning (Tibshirani, 1996; Chen et al., 1999; Peeters and Westra, 2004), the mode $\hat{a}$ of the posterior $P(a|X,u)$ is found through convex optimisation (recall that the log posterior is concave), and $\hat{a}$ is treated as posterior estimate of a. $\hat{a}$ has the property that many components are exactly zero:[5] the vector is sparse as such. This is useful for applications which aim for such exact sparsity, say for reasons of algorithmic efficiency. In contrast, in the Bayesian case, the posterior mass of all exactly sparse a (at least one component exactly zero) is zero, because the posterior has a density w.r.t. Lebesgue measure.[6] Not even commonly used Bayesian estimates of a, such as posterior mean or median, are exactly sparse in general. From a Bayesian viewpoint this makes sense, since in the presence of finite data,

5. One can easily show that as $\sigma^2 \to 0$, no more than m components of $\hat{a}$ can be non-zero.
6. Spike-and-slab sparsity priors have been used which place point masses on zero. However, approximate inference for such a setting is very challenging. Such priors are certainly not log-concave distributions.

one should always have some remaining uncertainty in exact values of parameters. The role of a sparsity prior in our situation is not to force many parameter values exactly to zero, but rather to enforce a clear partition into a large set of parameters which are close to zero with high (posterior) probability, and a small set which have significant mass on large values. Interestingly, following this probabilistic notion of sparsity sometimes allows to uncover sparsity in parameters of higher order that are of real interest, which is missed by MAP approaches. Our findings in Section 7.2 are a nice example of this effect.

2.2 Gene Network Identification

Measuring m-RNA expression levels for many genes in parallel is affordable and widely done today using DNA micro-arrays (DeRisi et al., 1997). One goal of such efforts is to recover regulatory networks. For example, some genes may code for transcription factor proteins, which up-/down-regulate the expression of other genes. In an *active* approach to network recovery, the evolution of expression levels of n genes is modeled by a system of ordinary differential equations, which is linearised at its steady state:

$$\dot{x}(t) = Ax(t) - u(t) + \varepsilon(t), \tag{3}$$

where $x(t)$ is the deviation in expression from steady state, and $\varepsilon(t)$ is white noise. A is the system matrix, whose non-zero entries represent the edges of the network. $u(t)$ is an external control, allowing the active user to probe the unknown A. It is generally assumed that $u(t)$ is small enough not to drive the system out of its linearity region. Due to the noisy environment, it is typical to restrict controls to be constant, $u(t) \equiv u$, and to measure the new steady state $\lim_{t\to\infty} x(t)$ (Tegnér et al., 2003). Such disturbances may be implemented biologically using gene switches (Gardner et al., 2000), which puts further restrictions on allowable u.

The linear model of (1) captures this setup as follows. Suppose that m observations $D = \{x_i, u_i\}$ have been made, where u_i is an external control, and x_i is the corresponding difference between steady state expression levels of the perturbed and the unperturbed system. We write $U = (u_i)^T \in \mathbb{R}^{m,n}$, $X = (x_i)^T \in \mathbb{R}^{m,n}$. We have that $u_i \sim N(Ax_i, \sigma^2 I)$. If a_i is the transpose of the i-th row of A, this Gaussian likelihood decomposes into n factors, one for each a_i. If the coefficients of A are assumed to be independent Laplacian *a priori*, the posterior factorises accordingly:

$$P(A|D) = \prod_j P(a_j|D), \quad P(a_j|D) \propto N(U_{\cdot,j}|Xa_j, \sigma^2 I) \prod_i P(a_{j,i}).$$

Thus, we have n independent sparse linear models, on which inference is done separately.

Since biological experiments involving gene switches are expensive and time-consuming, a key requirement is to perform with as few data as possible, which is possible if biological prior knowledge is encoded in $P(A)$. Importantly, regulatory networks are observed to be sparsely connected, that is, plausible A are sparse, a property which is directly represented in the sparse linear model. A principled way of saving on the number of expensive experiments is *optimal design*, which in a special case of interest here boils down to the question: given the current posterior belief and a set of candidate controls u_*, which of these experiments renders most new information about A? Thus, a "value of information" is sought which can be computed for each candidate u_* *without* doing the corresponding experiment. Optimal design is well developed in classical and Bayesian statistics (Fedorov, 1972; Chaloner and Verdinelli, 1995; MacKay, 1991), and access to this methodology is a key motivation for developing a good inference approximation here.

2.3 Coding of Natural Images

A second application of the sparse linear model is concerned with linear coding of natural images (Olshausen and Field, 1997; Lewicki and Olshausen, 1999), with the aim of understanding properties of visual neurons in the brain. Before we describe the setup, it is important to point out what our motivation is here, since it deviates significantly from what is usually done in machine learning. One approach in theoretical neuroscience is to formulate principles which can be described reasonably simply in mathematical terms, so that certain phenomena observed in experiments emerge if *only these principles are followed*. Once such principles are established, one can think about neural mechanisms implementing them. Also, if different principles lead to the same observed phenomena, one can plan experiments to further discriminate between them. In machine learning, the problems are known, and methods are compared with the aim of finding the best one, using an evaluation score and methodology independent of the set of methods to ensure a fair comparison. If results are not much different across methods, the most efficient one is usually preferred. In theoretical neuroscience,[7] the outcomes are known, and simple "universal" principles to explain them are sought. Once a principle is suggested, the aim is to devise a method following that principle as closely as possible. If such a method can then successfully reproduce observed phenomena, the principle can be established. In the context here, we are interested in testing a hypothesis put forward by Lewicki and Olshausen (1999), which is formulated in Bayesian terms. We are not interested here in coding images in the best possible way, and certainly not in how to do this with the highest computational efficiency.

An image $u \in \mathbb{R}^m$ is modeled as $u = Xa + \varepsilon$, where the columns of X are codebook vectors, $a \in \mathbb{R}^n$ are basis coefficients, and $\varepsilon \sim N(0, \sigma^2 I)$ independently. Note that codebook vectors are also referred to as filters, or basis functions. A central assumption on a is sparsity, which is especially important in the underdetermined (or overcomplete) regime: $m < n$. The Bayesian approach via the sparse linear model (1) has been suggested by Lewicki and Olshausen (1999), where the average coding cost of images under the model is put forward as criterion for ranking different code matrices X. Their work aims to give a probabilistic interpretation to the findings of Olshausen and Field (1997). In a Bayesian nomenclature, the average coding cost is the negative log marginal likelihood $-\log P(D)$, where $P(D) = \prod_j P(u_j)$, $P(u_j) = \int P(u_j|a_j)P(a_j)\,da_j$, and differences of these for different X are log Bayes factors. In Section 5, we show how to obtain a good approximation to $-\log P(D)$ through EP, which can be minimised w.r.t. the code matrix X in a gradient-based way. This general idea is proposed by Lewicki and Olshausen (1999) as well, but they use a second-order (Laplace) approximation to $-\log P(D)$, which is not suitable in case of a Laplace prior.[8] In the earlier approach of Olshausen and Field (1997), the learning of X is driven by point estimates (or maximum a posteriori decoding), and a criterion different from the average coding cost is optimised. This ignores posterior uncertainty in the decodings, and requires additional renormalisation heuristics in order to learn a good code. Our approximation here implements the probabilistic hypothesis of Lewicki and Olshausen (1999) fairly accurately, and can therefore be used to analyse more closely which of the features found by Olshausen and Field (1997) are due to the minimisation of average coding cost, versus which may rather be caused by particular characteristics of their learning method. Note that maximisation of the marginal likelihood is an important empirical

7. Or, in fact, in most natural sciences, with the exception of Engineering and Computer Science.
8. The problem is that $\log P(a_j)$ is not differentiable at the posterior mode $\hat{a}_j$, so that the matrix B in Lewicki and Olshausen (1999) is not well-defined. See comments in Section 3.

Bayesian way of estimating free hyperparameters, and Bayes factors are routinely used to compare model setups, so our approximation will be useful in other applications of the sparse linear model as well.

One of the key questions in natural image modelling is: under which conditions do basis vectors emerge which are spatially oriented and localised, thus show properties which have been established for the receptive fields of certain visual neurons? Given that the hypothesis of Lewicki and Olshausen (1999) is taken for granted, the sparsity hypothesis can be tested using the sparse linear model. Interestingly, other conditions brought forward (such as non-negativity) can also be dealt with in principle using the linear model, with different priors on a. Technically, non-negativity can be implemented by "cutting off" (and renormalising) a given prior density, which amounts to replacing $P(a_i)$ by $2P(a_i)\mathrm{I}_{\{a_i \geq 0\}}$. Importantly, if $P(a_i)$ is log-concave, so is this modification, because $\log \mathrm{I}_{\{a_i \geq 0\}}$ is (generalised) concave. For example, "cutting off" the Laplace distribution results in the exponential distribution,[9] which has been used in the context of image modelling by Hojen-Sorensen et al. (2002). While exponential priors encode non-negativity and sparsity at the same time, a cut-off Gaussian $P(a_i) = 2N(a_i|0, \tilde{\tau}^{-2})\mathrm{I}_{\{a_i \geq 0\}}$ could be used to represent non-negativity alone.

2.4 Bayesian Compressive Sensing

There has been a lot of recent interest in signal processing in the problem of compressive sensing (Candès et al., 2006; Donoho, 2006). The idea is appealingly simple. Suppose a signal is measured and then transferred over some channel or stored on some media. The second step almost always includes lossy compression in practice, especially with signals such as images or sound, where the loss may not be perceivable. Many of today's codes are *sparse*: the signal is transformed one-to-one, after which many coefficients are close to zero. These coefficients are then set to zero, and are not transmitted or stored. The first sensing (or sampling) step is traditionally done in a way which does not lead to loss of information, say by relying on the Nyquist/Shannon sampling theorem. The question of compressive sensing is whether one can sample a signal in a more efficient, but lossy way, so that the loss is part of that one encountered through subsequent compression anyway. The main attractiveness is that if a lossy compression is used, compressive sensing does not add further losses.

Although maybe not phrased in that way by much of the existing work, this is a classical problem of experimental design. An approximate Bayesian variant of compressive sensing has been proposed by Ji and Carin (2007), using sparse Bayesian learning (Tipping, 2001) to approximate the inference. Most practical codes today are linear, in that $y = \Phi a$, where y is the signal (say, an image), Φ is the code matrix (say, a Wavelet transform), and a are the coding coefficients. The code is designed such that a is approximately sparse, in much the same sense as elaborated in Section 2.1. Typically, Φ is one-to-one, even unitary. We then measure the signal linearly, that is, obtain $u = Py + \varepsilon$, where P is a measurement matrix, u are the responses, and ε is noise due to measurement errors. Here, $P \in \mathbb{R}^{m,n}$ with $m < n$ (the savings promised by compressive sensing). If $X = P\Phi$, this is exactly the setup of the linear model (1). Furthermore, the sparsity of a is encoded via a Laplace prior, motivating the sparse linear model for compressive sensing.

The measurement matrix P can be designed at will, where we are possibly limited to certain parametric families, due to constraints from the measurement architecture or (for very large n)

9. For this reason, the Laplace distribution is sometimes called double exponential distribution.

computational tractability (see Section 6). Anyway, we can design P row by row through an instance of standard sequential experimental design described in Section 4. This has been proposed in Ji and Carin (2007). Moreover, we can try to optimise P *a priori* over a large database of signals from the domain of the application, in what turns out to be an interesting variant of the image coding problem of Section 2.3. Here, the image code Φ is fixed, but P is to be learned.

Another point in which our approach differs from much of the existing work on compressive sensing, has to do with the sparsity prior we employ. Namely, many theoretical results have been obtained under the assumption that the signal y can be *exactly* sparsely coded, in that most coefficients in the corresponding a are exactly zero. However, in many real-world applications, this may be too strict an assumption. For example, the Wavelet transform of an image is virtually never exactly sparse, but rather features the bi-separation characteristic discussed in Section 2.1: many coefficients are very close to zero, and a subsequent quantisation leads to an image visually indistinguishable from y. Our sparsity prior concentrates on the bi-separation characteristic, without enforcing exact sparseness, thus may be better suited to many compressive sensing applications than the requirement of exact sparsity.

Results from experiments with different variants of compressive sensing are in preparation (joint work with Hannes Nickisch) and will be presented in a later paper (Seeger and Nickisch, 2008).

3. Expectation Propagation for the Linear Model

Exact Bayesian inference is not analytically tractable for the sparse linear model. In this section, we show how to apply the recently proposed *expectation propagation* (EP) method (Minka, 2001b; Opper and Winther, 2000) to this problem, circumventing some caveats we have not seen being addressed before. We begin with a high-level description, filling in the details further below. In the case of EP for the sparse linear model, it turns out that some details concerning robustness are essential for obtaining a practically useful method.

In EP, we compute a Gaussian approximation $Q(a)$ to the posterior

$$P(a|D) \propto N(u|Xa, \sigma^2 I) P(a).$$

Here, the likelihood $N(u|Xa, \sigma^2 I)$ is Gaussian, and it is the non-Gaussian prior $P(a)$ which forces us to approximate Bayesian inference. Our restriction to Gaussian $Q(a)$ is primarily done for pragmatic reasons, since Bayesian computations such as marginalisation and conditioning can be done analytically in this family, using standard matrix operations which can be computed robustly and efficiently. However, in our case, the Gaussian approximation can be argued for more strongly than in many others. Namely, recall that $\log P(a)$ is concave (2). Since the likelihood is a Gaussian function of a, the true log posterior $\log P(a|D)$ is concave as well, thus has a single mode only.

If $P^{(0)}(a) := N(u|Xa, \sigma^2 I)$ is the Gaussian likelihood (1), the true posterior is

$$P(a|D) \propto P^{(0)}(a) \prod_i t_i(a_i), \quad t_i(a_i) = \frac{\tilde{\tau}}{2} e^{-\tilde{\tau}|a_i|}.$$

We refer to the t_i as *sites*, and to $P^{(0)}$ as *base measure*. Note that the latter is not in general normalisable.

In order to motivate EP, note that an optimal Gaussian posterior approximation $Q(a)$ (at least in our context here) would be obtained by setting its mean and covariance to the true posterior

statistics. However, this would require a n-dimensional non-Gaussian integration, which cannot at present be done tractably. However, we are able to compute *one-dimensional* integrals involving a single non-Gaussian site $t_i(a_i)$. EP makes use of this capability in an iterative fashion, in order to approximate the desired joint posterior moments. The EP posterior approximation has the form

$$Q(a) \propto P^{(0)}(a) \prod_i \tilde{t}_i(a_i),$$

where $\tilde{t}_i(a_i|b_i,\pi_i)$ are Gaussian factors. Formally, one gets from the intractable $P(a|D)$ to its Gaussian approximation $Q(a)$ by replacing each non-Gaussian $t_i(a_i)$ by a Gaussian counterpart $\tilde{t}_i(a_i)$. This formal replacement introduces *site parameters* $b, \pi \in \mathbb{R}^n$, and the EP algorithm is an iterative method for adjusting these in turn.

In a single EP update, b_i, π_i are adjusted, while leaving all other site parameters the same. Starting from the current Gaussian approximation Q, we compute the Gaussian *cavity distribution* $Q^{\backslash i} \propto Q\tilde{t}_i^{-1}$ by dividing out the *site approximation* $\tilde{t}_i(a_i)$, then the non-Gaussian *tilted distribution* $\hat{P}_i \propto Q^{\backslash i} t_i$ by multiplying in the true site $t_i(a_i)$ instead, finally we update b_i, π_i such that the new Q' has the same mean and covariance as $\hat{P}_i$. These single updates are iterated in some random ordering over the sites until convergence.[10] Thus, EP is inherently based on the idea of *moment matching*. In other words, Q' is chosen by minimising the relative entropy $\mathrm{D}[\hat{P}_i \,\|\, \cdot]$ over all Gaussians.

From an algorithmic viewpoint, several questions have to be addressed. First, how can we *represent* the Gaussian $Q(a)$, so that single EP updates are served well in terms of efficiency and robustness? We will see that a good representation has to allow for the rapid "random-access" extraction of marginals $Q(a_i)$, and we have to be able to efficiently and robustly update it after a change of b_i, π_i. Second, how can the mean and variance of the non-Gaussian $\hat{P}_i(a_i)$ be computed accurately? To address these questions, we need to introduce some notation and details.

Denote the family of unnormalised Gaussian measures by

$$N^U(z|b,P) := \exp\left(-\frac{1}{2} z^T P z + b^T z\right),$$

P being positive semidefinite. Then, $P^{(0)}(a) = N^U(a|\sigma^{-2}b^{(0)}, \sigma^{-2}\Pi^{(0)})$ with $\Pi^{(0)} = X^T X$, $b^{(0)} = X^T u$. The site approximations are $\tilde{t}_i(a_i) = N^U(a_i|\sigma^{-2}b_i, \sigma^{-2}\pi_i)$, so that Q is a Gaussian. In general applications of EP, the π_i can become negative, but this does not happen in the cases discussed in this paper. We will show in Section 3.5 that for *log-concave sites* t_i, all π_i remain nonnegative throughout the course of the EP algorithm.

Moreover, the reader may wonder why we restrict ourselves to $\tilde{t}_i(a_i)$, instead of allowing for general site approximations $\tilde{t}_i(a)$. Also, a careful reader may have noted that we are only concerned about marginal distributions $Q(a_i)$ and $\hat{P}_i(a_i)$ during an EP update at t_i. Importantly, all this does not come with a loss of generality, as is shown in Section 3.1.

We initialise the algorithm with $b = 0$ and $\pi = \varepsilon 1$, $\varepsilon > 0$. A useful heuristic is $\varepsilon = \tau^2/2$, making sure that $t_i(a_i)$ and $\tilde{t}_i(a_i)$ have the same variance initially. In the case of the sparse linear model, the implementation of EP is complicated in a fundamental way. If $m < n$ (underdetermined case), the base measure $P^{(0)}(a)$ is not normalisable, because $\Pi^{(0)} = X^T X$ is singular. It is easily seen that

10. To our knowledge, little is known in general about convergence properties of EP, even with log-concave sites. Empirically, we have never observed failure of convergence in the log-concave case, except for reasons of numerical instability (see Section 3.3.1). Obtaining a formal convergence proof in this case remains a very important point for future research.

if any of the $\pi_i = 0$, the resulting $Q(a)$ is (in general) not a proper Gaussian either, so we have to ensure that $\pi_i > 0$ at all times. If $m \ll n$, we would like to represent the posterior Q in a way which scales with m rather than n. We address these issues below in this section.

It is important to note that EP is not merely a local approximation, in that $\tilde{t}_i$ is somehow fitted to t_i locally. This would not be useful at all,[11] because posterior mean and covariance are shaped *jointly* by the non-Gaussian t_i and the coupled Gaussian base measure. Loosely speaking, the likelihood couples coefficients a_i, so that the intentions of the prior factors $t_i(a_i)$, namely to force their respective arguments towards zero, have to be weighted against each other in a very non-local procedure.[12] After each EP update, although only a single site approximation is modified, its influence propagates to all other sites, because they are coupled through the base measure. In fact, non-locality is a central aspect of Bayesian inference which makes it so hard to compute, and inference is particularly hard to do in models where strong long-range posterior dependencies are present which cannot easily be predicted from local interactions only.

Finally, would it not be much simpler and more efficient to locate the true posterior mode through convex optimisation (recall that the posterior is log-concave), then do a Laplace approximation there, which amounts to expanding the log posterior density to second order around the mode? Indeed, finding the mode can be done efficiently by solving a quadratic program (Tibshirani, 1996). General problems with this approach include that the curvature around the mode may not be characteristic of the target density, and that the mode may not be a good place to center a Gaussian approximation at. In the case of the sparse linear model, the Laplace approximation is not even a valid option, since it is not well-defined in the presence of a Laplace prior.[13] Namely, $\log P(a_i)$ does not have a curvature at $a_i = 0$. The posterior mode is guaranteed to contain at least some zero components, so the curvature there is not defined. EP does not require $P(a_i)$ or $\log P(a_i)$ to be differentiable. On models where both methods can be applied, EP tends to improve upon a Laplace approximation significantly, but is also typically more expensive (Minka, 2001a; Kuss and Rasmussen, 2005).

3.1 Overview of Algorithm

In this section, we provide a schematic overview of the EP algorithm, filling in details in the sections to come. Recall that EP iterates site updates at $i \in \{1,\ldots,n\}$, computing $Q^{\backslash i} \propto Q\tilde{t}_i^{-1}$ and $\hat{P}_i \propto Q^{\backslash i} t_i$, then adjusting $Q \to Q'$ such that Q' has the same mean and covariance as $\hat{P}_i$. Since t_i depends on a_i only, $\hat{P}_i(a_{\backslash i}|a_i) = Q^{\backslash i}(a_{\backslash i}|a_i)$, where $a_{\backslash i} := (a_j)_{j\neq i}$, thus $Q'(a_{\backslash i}|a_i) = Q^{\backslash i}(a_{\backslash i}|a_i)$. Therefore, an EP update automatically results in the site approximation $\tilde{t}_i$ being a (Gaussian) function of a_i only. It also implies that in order to drive the EP update, all we need is the *marginal* distribution $Q(a_i)$. Just as most other variational "message-passing" approximate inference methods, EP can be seen as an iterative algorithm, improving estimates of the marginals $Q(a_i)$, $i = 1,\ldots,n$ until convergence. An EP update is *local*, in that its input is a marginal $Q(a_i)$ and it affects single site parameters b_i, π_i only. However, this *globally* affects all other marginals, which have to be updated through Gaussian propagation.

In common variational algorithms applied to discrete structured graphical models, such corrections of marginal estimates are performed by passing messages along the graph. In our case, the

11. Our experiments comparing Laplace and Gaussian priors in Section 7.1 illustrate this fact very nicely.
12. Our arguments about locality assume that a neighborhood structure can be imposed on a, say neighboring pixels in an image.
13. It is not known whether P. S. Laplace thought about this problem or even fixed it.

Algorithm 1 Expectation propagation algorithm for sparse linear model.

Require: X, u, τ, σ^2, η
  $b = 0$. $\pi = \varepsilon 1$. Compute initial representation of Q
  repeat
    for $i \in \{1, \ldots, n\}$ (random order) **do**
      Compute marginal $Q(a_i) = N(a_i | h_i, \sigma^2 \rho_i)$ from representation
      Do (fractional) EP update: $(b_i, \pi_i) \to (b_i', \pi_i')$
      Update representation of Q
    end for
    Refresh representation
  until marginal estimates $\{Q(a_i)\}$ converged

fully coupled Gaussian factor $P^{(0)}$ plays the role of the graph, and the messages are replaced by a *posterior representation* of $Q(a) = N(a|h, \sigma^2\Sigma)$. Just as with messages, the purpose of a representation is twofold: first, it needs to deliver mean and variance of an arbitrary marginal $Q(a_i)$ rapidly. Second, we need to be able to update it efficiently after each EP update. Our representations are given in Section 3.2, together with efficient update rules. Numerical errors can accumulate after many updates, so the representation is *refreshed* (i.e., recomputed from scratch) after each $O(n)$ EP updates. An iteration of EP updates over all (or most of the) sites is referred to as *sweep*. The structure of the EP approximate inference algorithm is given in Algorithm 1.

We close this section by remarking on the stopping rule we use in our EP implementation. One could stop once the site parameters do not change significantly anymore. However, we are really interested in the marginal means and variances, which in some cases are only weakly dependent on certain site parameters. For example, a large π_i means in general that the corresponding marginal mean is nailed down with a small variance, and increasing π_i further may have no large effect on the marginal distribution. Let $d(a,b) := |a-b| / \max\{|a|, |b|, 10^{-3}\}$ and $\Delta_i = \max\{d(h_i', h_i), \sigma d(\sqrt{\rho_i'}, \sqrt{\rho_i})\}$, where $Q(u_i) = N(h_i, \sigma^2\rho_i)$ and $Q'(u_i) = N(h_i', \sigma^2\rho_i')$ are the posterior marginals before and after an update at site i. We stop once $\max_i \Delta_i$ for a sweep over all sites is below some threshold.

3.2 Posterior Representation

In this section, we develop a representation of the posterior approximation $Q(a) = N(h, \sigma^2\Sigma)$ which allows efficient access to entries of h, $\operatorname{diag}\Sigma$ (marginal moments), and which can be updated robustly and efficiently for single site parameter changes (after EP updates). In fact, we propose two different representations: a degenerate and a non-degenerate one. The former is only useful in the underdetermined case ($m < n$), its updates are less numerically stable and more complicated, but it scales as $O(m^2)$, while the non-degenerate one scales as $O(n^2)$. If $m \ll n$, the degenerate representation leads to large computational savings.

We begin with the simpler non-degenerate representation:

$$\Sigma^{-1} = X^T X + \Pi = LL^T, \quad \gamma := L^{-1}(b^{(0)} + b),$$

where $\Pi := \operatorname{diag}\pi$ here and elsewhere. $L \in \mathbb{R}^{n,n}$ is the lower-triangular Cholesky factor (Horn and Johnson, 1985). Recall that $b^{(0)} = X^T u$. Note that $h = L^{-T}\gamma$. The marginal $Q(a_i) = N(h_i, \sigma^2\rho_i)$

is determined as $h_i = v^T\gamma$, $\rho_i = \|v\|^2$, where $v = L^{-1}\delta_i$. Here, δ_i is the Dirac unit vector with 1 at position i, and 0 elsewhere. This costs $O(n^2)$ (single back-substitution). After an EP update $b_i \to b_i'$, $\pi_i \to \pi_i'$, we have that

$$L'(L')^T = LL^T + (\pi_i' - \pi_i)\delta_i\delta_i^T, \quad L'\gamma' = L\gamma + (b_i' - b_i)\delta_i.$$

L', γ' are computed from L, γ using a Cholesky rank one update (downdate) for positive (negative) $\pi_i' - \pi_i$. This can be done in $O(n^2)$, we use a modification of the `LINPACK` routines `dchud`, `dchdd` (Dongarra et al., 1979), see Seeger (2004) for details. The update (downdate) is not done if $|\pi_i' - \pi_i|$ is too small. The reader may wonder why we do not represent and update Σ directly, using the Woodbury formula (see below). However, this would be numerically less stable than the Cholesky representation suggested here, and the operation count is the about the same.

In the underdetermined case $m < n$, another *degenerate* representation can be used, which leads to large savings if $m \ll n$. We noted in Section 3 above that Q is well-defined only if all $\pi_i > 0$. For numerical stability (with the degenerate representation), we require that $\pi_i \geq \kappa$ at all times, where $\kappa > 0$ is a small constant (we use $\kappa = 10^{-8}$ presently). This constraint is enforced in all EP updates. We can use the Woodbury formula (Henderson and Searle, 1981) in order to write

$$\begin{aligned}\Sigma &= \left(X^T X + \Pi\right)^{-1} \\ &= \Pi^{-1} - \Pi^{-1}X^T\left(I + X\Pi^{-1}X^T\right)^{-1}X\Pi^{-1}.\end{aligned}$$

We represent this via the lower-triangular Cholesky factor L in

$$LL^T = I + X\Pi^{-1}X^T.$$

Furthermore, let $\gamma := L^{-1}X\Pi^{-1}(b^{(0)} + b)$, whence

$$h = \Sigma(b^{(0)} + b) = \Pi^{-1}\left(b^{(0)} + b - X^T L^{-T}\gamma\right),$$

thus both h and Σ are represented by L, γ. For not too small κ, this representation is numerically stable. The marginal $Q(a_i)$ is obtained as $\rho_i = \pi_i^{-1}(1 - \pi_i^{-1}\|v\|^2)$, $h_i = \pi_i^{-1}(b_i^{(0)} + b_i - v^T\gamma)$, where $v := L^{-1}x$ with $x = X_{\cdot,i}$. After an EP update $b_i \to b_i'$, $\pi_i \to \pi_i'$, the representation is modified as follows. Let $\Delta_1 := (b_i^{(0)} + b_i')/\pi_i' - (b_i^{(0)} + b_i)/\pi_i$, $\Delta_2 := (\pi_i')^{-1} - \pi_i^{-1}$. We have that

$$L'(L')^T = LL^T + \Delta_2 xx^T, \quad L'\gamma' = L\gamma + \Delta_1 x.$$

Just as above, L', γ' can be computed from L, γ as a Cholesky rank one update/downdate, at the cost of $O(m^2)$. We do not modify π_i and the representation if $|\Delta_2|$ falls below some small threshold.

All in all, we can use a representation of Q whose size, as well as cost of a single site update, is quadratic in the smaller of n and m. Beware that L, γ have different definitions in the two cases. Note that we can also use the non-degenerate representation in the case $m < n$. In general, the non-degenerate representation leads to more numerically stable computations (supposedly because the Woodbury formula is not used), which are in fact more efficient in practice once $m \approx n/2$. We recommend to use the degenerate representation only if significant computational savings are observed in practice.

In some experimental design applications, such as gene network identification considered here, $m \ll n$ initially, but m grows up to $n/2$ eventually. In such cases, one could be tempted to use the degenerate representation initially, then switch to the non-degenerate one. In general, this does not make sense, since the majority of the computational effort is spent in the later stages anyway, and the non-degenerate representation should be used throughout.

3.3 The EP Update

An EP update works by matching moments between a tilted and the new posterior distribution. For an update at site i, we require the marginal $Q(a_i) = N(h_i, \sigma^2\rho_i)$ only, which is obtained from the Q representation. The moment matching requires the computation of Gaussian expectations with $t_i(a_i)$, a univariate quadrature which in general is not an analytical computation.

If $Q^{\backslash i}(a_i) = N(h_{\backslash i}, \sigma^2\rho_{\backslash i})$, we have that

$$\rho_{\backslash i} = \frac{\rho_i}{1-\rho_i\pi_i}, \quad h_{\backslash i} = \frac{h_i - \rho_i b_i}{1-\rho_i\pi_i}.$$

If the degenerate representation is used, it is more stable to compute the cavity marginal directly. Namely, if $v := L^{-1}X_{\cdot,i}$, then $\rho_{\backslash i} = \|v\|^{-2} - \pi_i^{-1}$ and $h_{\backslash i} = (b_i^{(0)} - v^T\gamma)/\|v\|^2 + b_i/\pi_i$.

Next, we need to compute mean and variance of $\hat{P}_i(a_i) = Z_i^{-1}Q^{\backslash i}(a_i)t_i(a_i)$, which we do as described in Seeger (2003), Appendix C.1.3. Note that $Z_i = \mathrm{E}_{Q^{\backslash i}}[t_i(a_i)]$, and define $\beta_i := (d\log Z_i)/(dh_{\backslash i})$, $\nu_i := -(d^2\log Z_i)/(dh_{\backslash i}^2)$. The concrete computation of β_i, ν_i (or equivalently, of the first and second moment of $\hat{P}_i(a_i)$) can be done analytically for Laplace sites, but is not straightforward due to issues of numerical stability, it is described in Appendix A. Then, the new site parameters are given by

$$\pi_i' = \frac{\sigma^2\nu_i}{1-\sigma^2\nu_i\rho_{\backslash i}}, \quad b_i' = \frac{\sigma^2(\beta_i + h_{\backslash i}\nu_i)}{1-\sigma^2\nu_i\rho_{\backslash i}}.$$

We show in Section 3.5 that $\nu_i \geq 0$, thus $\pi_i' \geq 0$, due to the log-concavity of t_i. If $\pi_i' < \kappa$ and the degenerate representation is used, we set $\pi_i' = \kappa$.

The numerical difficulties with the EP update for Laplace sites are remarkable, given that no such problems occur in several other EP applications, for example Gaussian process classification (GPC) with probit or logit noise (Minka, 2001b; Opper and Winther, 2000; Lawrence et al., 2003), where less careful implementations still work fine, and even approximate Gaussian quadrature can be used. Several early attempts of ours led to complete failure of the algorithm on realistic data (in the underdetermined case), motivating the fairly elaborate solution in Appendix A. While we cannot offer a firm explanation for this yet, our intuition is that the effect of Laplace prior sites on the posterior is much stronger, trying to emulate the essentially discrete feature selection process in a "unimodal" manner. Our findings also shed some sceptical light on proposals to implement a generic toolbox for EP, applying Gaussian quadrature[14] to do EP updates for general sites (Zoeter and Heskes, 2005). In the gene network identification application, we ran into problems of numerical instability coming from the combination of Laplace sites with very underdetermined coupling factors $P^{(0)}$. We suspect these problems are inherent, and in our case could be handled only by considering a modification of EP, as discussed just below.

3.3.1 FRACTIONAL EP UPDATES

We just mentioned the numerical difficulty of doing EP updates with Laplace sites in the strongly underdetermined case $m < n$. A frequent cause of numerical problems with EP is sloppiness in

14. Gaussian quadrature would fail completely for sites like the Laplace, which are not smooth functions. A central assumption with virtually all quadrature methods today is that the integrand up to a predefined weight function can be closely approximated by a low-order polynomial. Note that Monte Carlo integration is usually not considered useful for (low-dimensional) quadrature, due to its poor relative accuracy.

the implementation. For example, representation updates based on the Woodbury formula are a frequent source of accumulation of round-off and cancellation errors (see Section 3.2). The EP update with Laplace sites is quite difficult to do in a stable way (see Appendix A).[15] However, even using all these careful measures did not allow us to run standard EP on many of the gene network identification problems of Section 7.1 or on a fraction of the image coding problems of Section 7.2. We think that these stability problems of EP are inherent for some tasks, giving some motivation below. Fortunately, EP can be modified to use fractional updates, which in fact counter exactly the numerical problems we face. While fractional EP has been suggested as alternative to standard EP (Minka, 2004), its role for circumventing stability problems has not been noted so far to our knowledge.

Recall from Section 3 that if we set all or most of the $\pi_i = 0$ in the underdetermined case, the variance of most marginals $Q(a_i)$ is infinite. We face this problem by ensuring that $\pi_i \geq \kappa$ at all times. Still, at least for some updates, the cavity marginal variance of $Q^{\setminus i}(a_i)$ is huge. This is because we divide through the site approximation $\tilde{t}_i(a_i)$, whose $\pi_i \geq \kappa$ keeps the variance small. The variance is not infinite due to the effect of the other $\pi_j \geq \kappa$ and the coupling through $P^{(0)}$, but in many underdetermined situations, this coupling is weak. We then try to do an EP update based on a very wide cavity distribution $Q^{\setminus i}(a_i)$ and a quite narrow site $t_i(a_i)$ (enforcing a strong sparsity constraint requires a rather large τ). This is inherently difficult to do.

It would be better to make $Q^{\setminus i}(a_i)$ narrower and $t_i(a_i)$ wider, which is exactly what happens in *fractional EP updates*. Here, we obtain $Q^{\setminus i}(a_i)$ by dividing out only a fraction of $\tilde{t}_i(a_i)$, and $\hat{P}_i(a_i)$ by multiplying with only a fraction of $t_i(a_i)$. This idea is fairly natural, simply imagine the sites being replicated q times, then taken to the power of $\eta = 1/q$ to obtain the original setup. The only difference to standard EP is that we tie the parameters of the corresponding fractional site approximation replicas. Of course, the idea is not limited to rational fractions. Some extensions and theory of this method are discussed by Minka (2004). Another view on fractional EP is that projections from standard EP's $\hat{P}_i$ to Q' are done based not on the relative entropy (see Section 3), but on an α-divergence depending on the fraction.

For the fraction parameter $\eta \in (0,1]$, let $Q^{\setminus i} \propto Q\tilde{t}_i^{-\eta}$ and $\hat{P}_i \propto Q^{\setminus i} t_i^{\eta}$. We choose the new site parameters b_i', π_i' such that the moments of $\hat{P}_i$ and Q' match. This can be incorporated into the derivations above by setting $\tilde{b}_i = \eta b_i$, $\tilde{\pi}_i = \eta\pi_i$, and $\tilde{\tau} = \eta\tau$. The cavity moments are computed as

$$\rho_{\setminus i} = \frac{\rho_i}{1 - \rho_i \eta \pi_i}, \quad h_{\setminus i} = \frac{h_i - \rho_i \eta b_i}{1 - \rho_i \eta \pi_i}.$$

For the degenerate representation, a direct computation may be more stable:

$$\rho_{\setminus i} = \pi_i^{-1} \frac{R}{1 - \eta R}, \quad h_{\setminus i} = \pi_i^{-1} \left(\frac{b_i^{(0)} - v^T \gamma}{1 - \eta R} + b_i \right), \quad R = 1 - \pi_i^{-1} \|v\|^2.$$

We then compute $\tilde{b}_i', \tilde{\pi}_i'$ as above, using $\tilde{\tau} = \eta\tau$ instead of τ in the Laplace site, so that $\hat{P}_i$ and $\propto Q^{\setminus i}\tilde{t}_i(\cdot|\tilde{b}_i', \tilde{\pi}_i')$ have the same moments. Fractional updates are easily implemented for sites $t_i(a_i|\tau)$ with some hyperparameter τ, such that $t_i(a_i|\tau)^{\eta} = t_i(a_i|\eta\tau)$. The Laplace site is of this kind, if the normalisation constant of $\tau/(2\sigma)$ is dropped (it does not affect mean or variance of $\hat{P}_i$). Note that in

15. It is even harder to do for certain non-log-concave sites. For example, the sparse linear model with Student's t prior would be very hard to address with standard EP (Malte Kuss, pers. comm.).

general, t_i^η is log-concave if t_i is. Finally, the site parameters are updated as

$$b_i' = (1-\eta)b_i + \tilde{b}_i', \quad \pi_i' = (1-\eta)\pi_i + \tilde{\pi}_i',$$

upon which $\hat{P}_i$ and the new Q' have the same moments.

Another idea of making EP run smoother on hard problems is *damping* (Minka, 2001a). There, the full standard EP update is computed, but the site parameters are updated to a convex combination of old and proposed new values. This addresses a quite contrary problem to ours here. Damping is useful if EP update computations are stable, but lead to an improper new posterior, or the propagation of the updated information fails. If EP is viewed as finding a saddle point of a free energy approximation (Opper and Winther, 2005), damping can be understood as a step-size rule within this process. It slows down convergence in general in situations where EP without damping works fine, but the fixed points are not altered. Our problem is not solved by damping, since proposed new values for the site parameters cannot even be computed.

Finally, the reader may wonder whether the problems with standard EP are due to a bad initialisation of the site parameters. While we have not analysed it in all details, we think the problem is inherent. For example, we tried to run fractional EP to convergence, then start standard EP (with $\eta = 1$) from the fractional fixed point. On critical cases, this fails about as fast as if started in the usual way, often in the first sweep of standard EP.

3.4 Inclusion of a New Point

Suppose we would like to operate inference in the sparse linear model in a sequential manner, in that new data points (x_*, u_*) become available over time. This is the case in sequential design applications, since single experiments result in new measurements. In this section, we show how the EP posterior representation is updated once a new point (x_*, u_*) is added to the current data set D. The inclusion of (x_*, u_*) works in two stages. First, the Gaussian base measure is modified in order to incorporate the new point. Second, EP updates are done until convergence. The mechanics of the latter have been described above, so we can concentrate on the first stage here.

For the non-degenerate representation, let $v := L^{-1}x_*$. The change of $b^{(0)}$ results in $\tilde{\gamma} = \gamma + u_* v$. Since $L'(L')^T = LL^T + x_* x_*^T$, L', γ' is obtained from $L, \tilde{\gamma}$ by a rank one Cholesky update (see Section 3.2). The cost is $O(n^2)$.

For the degenerate representation, let $X' = (X^T, x_*)^T \in \mathbb{R}^{m+1,n}$ and $u' = (u^T, u_*)^T \in \mathbb{R}^{m+1}$. Since $b^{(0)} = X^T u$, we have that $b^{(0)\prime} = b^{(0)} + u_* x_*$. Let $l := L^{-1} X \Pi^{-1} x_*$. Then, $\tilde{\gamma} = \gamma + u_* l$ incorporates the update of $b^{(0)}$. Next, LL^T grows by a row/column $((Ll)^T, 1 + x_*^T \Pi^{-1} x_*)^T$. Therefore, L', γ' are obtained from $L, \tilde{\gamma}$ by a Cholesky extension, as described in Seeger (2004). The cost of the inclusion is $O(m^2)$.

3.5 Some Consequences of Log-concavity

A nonnegative function $f(x)$ is *log-concave* if

$$f(\lambda x_1 + (1-\lambda)x_2) \geq f(x_1)^\lambda f(x_2)^{1-\lambda}$$

for all x_1, x_2, and $\lambda \in [0,1]$. $f(x)$ is log-concave iff $\log f(x)$ is concave as a generalised function, which can take on the value $-\infty$, see Boyd and Vandenberghe (2002), Sect. 3.5. We call a distribution log-concave, if its density exists and is log-concave. In this section, we show some implications of log-concave sites for the numerical stability of EP.

Gaussians are clearly log-concave, so models of the sort considered here are log-concave if the sites are (products of log-concave functions are log-concave). For example, Laplace sites $t_i(a_i)$ are log-concave, while Student's t sites are not. A direct consequence is that for log-concave sites, the posterior is log-concave, so its unique mode can be found by convex optimisation. Log-concavity is stronger than unimodality though. For example, all upper level sets (areas enclosed by contours) of the posterior are convex sets. Intuitively, log-concave distributions are "simple", although strong consequences of this fact for variational approximate inference methods are not known to our knowledge.[16] Our main result is the following theorem.

Theorem 1 *Let EP be applied to a model with true posterior of the form*

$$P(a|D) \propto P^{(0)}(a) \prod_i t_i(a_i),$$

where $P^{(0)}(a)$ is a joint unnormalised Gaussian factor, and the sites $t_i(a_i)$ are log-concave. Suppose the site parameters π_i are initialised to non-negative values. Then, all EP updates are computable (in exact arithmetic), and all π_i remain non-negative throughout.

The proof is given in Appendix A.1. The theorem holds just as well for general sites $t_i(a)$ with corresponding site approximations $\tilde{t}_i(a) = N^U(\sigma^{-2}b_i, \sigma^{-2}\Pi_i)$, if "$\pi_i \geq 0$" is replaced by "$\Pi_i$ positive semidefinite". It hinges on a fundamental marginalisation theorem for log-concave functions due to Prékopa, see Bogachev (1998). Namely, suppose that $f(x,y)$ is jointly log-concave in (x,y), $x \in \mathbb{R}^p$, $y \in \mathbb{R}^q$. Then $\int f(x,y)\,dy$ is log-concave in x. Theorem 1 implies that EP can be implemented in a numerically stable way. Namely, the non-negativity of all π_i ensures that the representations introduced in Section 3.2 can be updated in a stable manner. The situation for some applications with non-log-concave sites is much less satisfactory. It is usually not possible to keep all π_i positive anymore, without making significant approximation errors (Minka, 2001a). Full EP updates lead to erratic behaviour or cannot even be done, and damping has to be used, leading to slower convergence. Negative entries π_i can lead to very ill-conditioned Cholesky factors in the representations, resulting in large errors at each update.

Our theorem also implies that for applications where EP is started with $\pi = 0$, for example Gaussian process classification, we have that the entropy $\mathrm{H}[Q]$ of the posterior decreases monotonically during the first sweep. Namely, the entropy is $\log|\Sigma|$ up to constants, which is decreasing in every single π_i. Minka (2001a) notes that the first sweep of EP is equivalent to a method called *assumed density filtering* (Kushner and Budhiraja, 2000), so our theorem has implications for this method as well.[17]

Another interesting consequence of log-concavity holds for the sparse linear model, independent of whether EP is used for approximate inference or not. It serves to motivate the parameterisation of the Laplace sites (2) in terms of $\tilde{\tau} = \tau/\sigma$. Up to additive constants, $\log P(u,a)$ has the form

$$(2m+n)\log\sigma^{-1} - \frac{1}{2}\|u/\sigma - Xa/\sigma\|^2 - \tau\sum_i |a_i/\sigma|,$$

16. In contrast, MCMC sampling from log-concave distributions has been proven to be computationally efficient (Lovász and Vempala, 2003).

17. The entropy $\mathrm{H}[Q]$ can increase in later sweeps of EP (this happens regularly, not only in special cases). This is why we need to consider Cholesky *downdates* in Section 3.2, and shows that $\mathrm{H}[Q]$ alone cannot be used to prove convergence of EP.

which is jointly concave in (ϕ, σ^{-1}), where $\phi := a/\sigma$. This fact has been noted in Park and Casella (2005). In fact, even $P(u,a)\sigma$ is log-concave in (ϕ, σ^{-1}), since $2m+n \geq 1$. The marginal likelihood $P(u)$ is a crucial criterion when it comes to hyperparameter optimisation or Bayesian tests (see Section 5). Now,

$$P(u) = \int P(u,a)\, da = \int P(u,a)\sigma\, d\phi,$$

and by the marginalisation theorem, $P(u)$ is log-concave in σ^{-1}. This implies that if all other hyperparameters are fixed, the empirical Bayesian maximisation of $\log P(u)$ w.r.t. the noise variance σ^2 is in fact a convex problem with a unique solution. Unfortunately, this property does not extend to other hyperparameters such as τ or X. On a practitioner's level, it is interesting to relate this fact to a scheme mapping out the entire regularisation path of Lasso (or, equivalently, an SVM) (Hastie et al., 2004). In either case, adjusting one hyperparameter trading off prior and likelihood given all others is shown to be simple. Here, as there, this gives some reassurance if σ^2 is adapted along with other parameters (see Section 7.2).

We close this section by some technical side comments for readers interested in details, all others may skip this paragraph. We require results from Section 5. We just showed that the exact $\log P(u)$ is concave in σ^{-1}, but how about the EP approximation of this quantity, called L in Section 5? To answer this question, we first have to establish that L is well-defined and continuous as a function of σ^2 in the first place. Now, L is defined in terms of the site parameters at convergence, and the EP algorithm has not been proven to always converge uniquely. Opper and Winther (2005) show that L is a proper approximate free energy as function of b, π, but the site parameters at convergence are only a saddle point thereof. Using tools such as the implicit function theorem, one can argue that $L(\sigma^2)$ is well-defined across some range, but does this hold globally across all σ^2? If the dependence of the site parameters on σ^2 is ignored locally, then L is log-concave in σ^{-1}, following similar arguments as above. We know from Section 5 that for computing the first derivative w.r.t. σ^2, the site parameters can be assumed constant, but this is not true in general for the second derivative (which would characterise concavity). Clearly, there is more work needed to gain a better understanding of such properties of the implicitly defined EP approximate free energy L.

4. Sequential Optimal Design

The role of sequential optimal design[18] for saving on expensive experiments has already been motivated in Section 2. The topic is well-researched in classical and Bayesian statistics (Fedorov, 1972; Chaloner and Verdinelli, 1995). A variant is known in machine learning as *active learning*[19] (Seung et al., 1992). We follow MacKay (1991) here, whose setting is closest to ours.

In the sparse linear model, a typical design problem can be formulated as follows. Given a set of candidate points x_*, at which of these should a corresponding target value u_* be sampled in order to obtain as much new information about the unknown a as possible? Assuming (for the moment) that u_* is known for a x_*, natural scores quantify the decrease in posterior uncertainty or gain in

18. *Optimal design* is a fixed term in statistics for a methodology, in which designs are optimised. We have no intention of claiming that any of the methods presented here solve problems in an optimal way, in fact they usually do not. In the context of this paper, *optimal design* and *experimental design* mean the same thing.
19. Confusingly, *active learning* is also used for the related, but not identical setup, where data comes in sequentially, and the method has to decide which cases to incorporate versus which to ignore. We are not interested here in this latter setting.

information from the current posterior Q to the novel Q' which is obtained by including (x_*, u_*) into the data set D. In this paper, we concentrate on the information gain $\mathrm{D}[Q' \,\|\, Q] = \mathrm{E}_{Q'}[\log Q' - \log Q]$. A large information gain means that Q' is different from Q, thus much novel information is gained from (x_*, u_*). Now, u_* is *not* known for the candidates x_*. Bayesian methodology dictates that u_* is averaged over its current posterior $Q(u_*|x_*, D) = \int P(u_*|x_*, a) Q(a|D)\, da$. Since $Q(a|D)$ is Gaussian for our approximation, the posterior over u_* is Gaussian as well. A natural score for x_* is the expected information gain $\mathrm{E}_{Q(u_*|x_*,D)}[\mathrm{D}[Q' \,\|\, Q]]$. This one-dimensional integral can easily be approximated using Gaussian quadrature.

However, optimal design for the gene network application of Section 2 does not fall into this standard category and requires some additional thoughts. The goal is to score the utility of inclusion of candidate controls u_*, given current data D (and posterior Q). Among a list of candidates, the highest-scoring u_* is then subjected to a new experiment in order to obtain x_*, whence (u_*, x_*) are included to form $D' = D \cup \{(u_*, x_*)\}$ and a new posterior Q'. Here, the posterior is a product of independent factors for the rows of A, so that for given (x_*, u_*), the information gain is the sum of $\mathrm{D}[Q' \,\|\, Q]$ over the posterior factors, where $(x_*, u_{*,j})$ is appended to D for the j-th factor.

More importantly, it is x_* which is unknown, rather than u_* in the standard setup. While $Q(u_*|x_*, D)$ is a Gaussian in our setup, $Q(x_*|u_*, D) = \int P(x_*|u_*, A) Q(A|D)\, dA$ is not a simple distribution. However, we can easily sample from it by first drawing $A \sim Q(A|D)$, then[20] $x_* = A^{-1}(u_* - \varepsilon)$, $\varepsilon \sim N(0, \sigma^2 I)$. Sampling from $Q(A|D)$ is discussed in Appendix B.2. Our *information gain* score in the gene network application is

$$S(u_*; D) = \mathrm{E}_{Q(x_*|u_*,D)} \left[\mathrm{D}[Q' \,\|\, Q] \right],$$

where the expectation is approximated by using a number of independent samples x_*.

Going back to the standard setup, for fixed (u_*, x_*), Q' is obtained from the current Q by first modifying the base measure $P^{(0)}$ corresponding to the inclusion, then updating the site parameters b, π. The problem with this is that the EP updates are expensive, so only few candidates could be scored for each inclusion.[21] A simpler and much cheaper alternative is to *approximate* the information gain by modifying $P^{(0)}$ only, but keeping the old site parameters, when defining Q' in $\mathrm{D}[Q' \,\|\, Q]$. In other words, for the purpose of scoring, we treat the model as purely linear-Gaussian, with Q as "effective Gaussian prior". This simple score can be computed very efficiently and reliably, so many candidates can be scored. Details are given in Section 4.1. Recall that the score for the gene network setup is the sum of information gains for the posteriors of each row of A.

Once a candidate is chosen for inclusion, a true experiment is done in order to obtain a complete new data point. In the standard setup, this means drawing u_*, given x_*, but in the gene network setting, we determine x_* for given control u_*. The new information is then included by a posterior update, as described in Section 3.4, and the site parameters are driven to new convergence by the EP algorithm.

Note that the fact that we approximate the true posterior $P(a|D)$ by a Gaussian $Q(a)$, as well as use Q' with the same site parameters as Q, means that we merely approximate the information gain, and at present we cannot give useful approximation guarantees, beyond our empirical demonstrations that good designs are usually found. However, our use of Gaussian Q and a simple update

20. We use a LU decomposition of A. The cost of $O(n^3)$ may be prohibitive for large n (although the same A sample can be used to score *all* candidates), in which case we would recommend sparsifying A and using a sparse matrix solver.
21. One idea would be to update few sites only after each inclusion. The extension described in Section 6.3 could be used to implement this, which is however not done here.

$Q \to Q'$ means that our information gain approximation can be computed robustly, which, as noted in Section 1, is often as important for experimental design as is high approximation accuracy.

4.1 Simple Information Gain

The simple information gain score is $\mathrm{D}[Q' \,\|\, Q]$, where Q is the current posterior, and Q' is obtained from Q by including (x_*, u_*) into the base measure $P^{(0)}$ as discussed in Section 3.4, but leaving the site parameters b, π at their old value. The relative entropy between Gaussians is well known:

$$\begin{aligned}\mathrm{D}[N(h', \sigma^2\Sigma') \,\|\, N(h, \sigma^2\Sigma)] = &\frac{1}{2}\log|M| + \frac{1}{2}\mathrm{tr}\left(M^{-1} - I\right) \\ &+ \frac{1}{2}\sigma^{-2}(h'-h)^T\Sigma^{-1}(h'-h), \quad M := (\Sigma')^{-1}\Sigma.\end{aligned} \tag{4}$$

Importantly, in our case we have that $(\Sigma')^{-1} = \Sigma^{-1} + x_* x_*^T$, so that $M = I + x_* x_*^T\Sigma$ has a simple form. This allows us to compute the simple information gain very efficiently. Details are given in Appendix B.1.

4.2 Marginal Criteria

The simple information gain scored discussed in the previous section measures the distance between the *joint* distributions Q and Q' (before and after inclusion). However, inference schemes such as expectation propagation (and other variational ones) are designed to approximate the posterior marginals well. EP applied to models with Gaussian base distribution results in a full joint posterior approximation Q, which can of course be used to make decisions or to compute information scores, but very little is known about the quality of Q beyond its marginals. A careful approach would therefore base experimental design scores on the marginals of Q only. On the other hand, criteria based on the full joint posterior can be more powerful in order to distinguish between many candidates.

For such marginal scores, we need to know how the marginals change after an update of $P^{(0)}$. Let $h, \mathrm{diag}\,\Sigma$ be the current marginal moments. We need to compute $h', \mathrm{diag}\,\Sigma'$ after inclusion of (x_*, u_*) (we only deal with the "simple" variant here, where no EP updates are done after the inclusion). Let $\alpha = x_*^T\Sigma x_*$, $z_* = \Sigma x_*$. By the Woodbury formula, we have that

$$\Sigma' = \Sigma - (1+\alpha)^{-1} z_* z_*^T.$$

The new h' is given in Appendix B.1, where we also show how to compute z_*, α efficiently. The marginal moments $h, \mathrm{diag}\,\Sigma$ have to be computed from the representation before each scoring round, although another idea is developed in Section 6.

Interestingly, in initial gene network identification experiments, employing the sum of marginal information gains worked less well than using the simple joint information gain of Section 4.1. In this case, the latter seems to carry more useful information about the candidates. In other words, the posterior correlations estimated by EP seem good enough to be useful here. Results with marginal scores are not reported in this paper.

5. The Marginal Likelihood

Bayesian methodology requires that unobserved variables are marginalised over in order to do predictions or to make optimal decisions. However, in many situations this is not practically feasible

for some variables. In the case of the sparse linear model, we can approximately integrate out the parameters a using EP, but likelihood and prior depend on other *hyperparameters* still, namely σ^2, τ, and (parameters of) X. A surrogate widely accepted amongst Bayesian practitioners is to *estimate* good values for the hyperparameters by maximising the *marginal likelihood* of the observed data.[22]

An approximation to the marginal likelihood may be obtained from EP. Details about this approximation can be found in Seeger (2005). As shown there, it is the same as the approximate free energy proposed in Opper and Winther (2005). We give the derivation for fractional EP in general (see Section 3.3.1), $\eta \in (0,1]$. Standard EP is obtained for $\eta = 1$. We have that

$$P(D) = P(u) = \int \prod_{i=1}^{n} t_i(a_i) P^{(0)}(a)\, da. \tag{5}$$

Recall that in EP, the sites t_i are replaced by approximations $\tilde{t}_i$ of Gaussian form. Earlier on, we did not bother with the normalisation constants of these approximations, but now we have to make them explicit: $t_i(a_i) \to C_i \tilde{t}_i(a_i)$, $\tilde{t}_i(a_i) = N^U(a_i|\sigma^{-2}b_i, \sigma^{-2}\pi_i)$. Roughly speaking, EP works by making the first and second order moments of the posterior marginals $Q(a_i)$ and the tilted distributions $\hat{P}_i(a_i)$ equal for all i. In this line, we fix the C_i such that the normalisation constants are the same as well:

$$\log C_i = \eta^{-1}(\log Z_i - \log \tilde{Z}_i), \quad Z_i = \mathrm{E}_{Q^{\setminus i}}\left[t_i(a_i)^\eta\right],\ \tilde{Z}_i = \mathrm{E}_{Q^{\setminus i}}\left[\tilde{t}_i(a_i)^\eta\right].$$

Here, $Q^{\setminus i} \propto Q\tilde{t}_i(a_i)^{-\eta}$. The EP approximation $L \approx \log P(u)$ is then obtained by replacing t_i by $C_i\tilde{t}_i$ in (5). This results in

$$L = \sum_{i=1}^{n} \log C_i + \frac{1}{2}\left(\log|\Sigma| + \sigma^{-2}h^T(b^{(0)} + b) - \sigma^{-2}\|u\|^2 + (n-m)\log(2\pi\sigma^2)\right). \tag{6}$$

In order to maximise L, we require its gradient w.r.t. hyperparameters, which can be computed exactly if EP is run to convergence, such that the moments of all $Q(a_i)$ and $\hat{P}_i(a_i)$ coincide. In Seeger (2005), the following is shown:

$$\nabla_{\theta^{(0)}} L = \mathrm{E}_Q[\nabla_{\theta^{(0)}} \log P^{(0)}(a)], \tag{7}$$

where $\theta^{(0)}$ are the natural parameters of $P^{(0)}$. Furthermore, if α is a parameter of the site t_i independent of $P^{(0)}$, then

$$\frac{\partial L}{\partial \alpha} = \frac{\partial \log Z_i}{\partial \alpha} = \mathrm{E}_{\hat{P}_i}\left[\frac{\partial}{\partial \alpha}\log t_i(a_i)\right]. \tag{8}$$

Note that this holds for fractional EP in general, if $\eta \in (0,1]$. The specialisations to our case are given in Appendix C.

Note that the EP approximation L of $\log P(u)$ has an important consistency property. It is well known that $\nabla_{\theta^{(0)}} \log P(u) = \mathrm{E}_{P(a|D)}[\nabla_{\theta^{(0)}} \log P^{(0)}(a)]$, from which (7) is obtained by replacing the true posterior by the EP approximation $Q(a)$: the true gradient of the approximate criterion is the approximate gradient of the true criterion. Another way to view this is to note that L depends on hyperparameters directly as well as through the EP site parameters b, π, thus the gradient has direct as well as indirect contributions. Importantly, the stationary conditions of EP at convergence

22. In work in progress, we show how the noise variance σ^2 can be integrated out along with a using EP. This, however, leads to a significantly more complicated and somewhat less robust algorithm. Details will be given in a later paper.

imply that the latter contributions do vanish. In Seeger (2005), it is shown that $(\partial L)/(\partial \sigma^{-2} b_i) = (\partial L)/(\partial \sigma^{-2} \pi_i) = 0$ at EP stationary points, which directly implies the simple formulae (7), (8). This fact becomes clear if L is seen as approximate free energy (Opper and Winther, 2005) (although only the case $\eta = 1$ is discussed there).

Note that the fraction η is a parameter of the approximation method, not a statistical variable or hyperparameter. It is natural to ask for which η one would obtain the best approximation of $\log P(u)$. Since L is not a bound on $\log P(u)$, we cannot directly optimise η. Useful theoretical insights about fractional EP variational free energies for different η are not known to us. We will address this point as part of an empirical comparative study, which is subject to future work. However, note that EP cannot be run at all in a stable way for certain setups if η is (very close to) one (see Section 7.1).

6. Large-Scale Applications

A naive implementation of the EP algorithm for the sparse linear model requires $O(n^3)$ time for each sweep and $O(n^2)$ memory, if the non-degenerate representation is used. While this is acceptable for moderate sizes of n, like in the gene network identification application, it is certainly not feasible for large n. In this section, we propose some ideas in order to apply our framework in such situations.

The dominant computation within our framework, both for experimental design and marginal likelihood maximisation, is spent performing a sweep of EP updates. Naively, this is a loop over all n sites (in random ordering). For each site i, the marginal $Q(a_i)$ has to be determined, and the representation for doing so has to be updated afterwards. Time can be saved by doing less than n updates per sweep, and by speeding up the marginal extraction or representation update. In a sequential context such as experimental design, it is sensible to assume that many EP updates will not lead to much change in Q, especially during later stages. A key problem is how to efficiently detect the sites whose update would change the current posterior the most.

Furthermore, a large-scale application will normally not be generic, but comes with a lot of structure already. For example, the design matrix $X \in \mathbb{R}^{m,n}$ is often given implicitly, since its storage alone would be too costly, and matrix-vector multiplications (MVMs) with X are often much more efficient than $O(nm)$. A key step in the direction of a large-scale implementation is to make sure that such special structure is used optimally.

A motivating example for a large-scale application comes from compressive sensing (see Section 2.4). Recall that $X = P\Phi$, where Φ is a fixed coding matrix, and P is a measurement matrix we can design. If the task deals with full images, n can be a million, and neither Φ nor P can be stored as dense matrices, but have to be defined implicitly as linear mappings. For example, Φ could be an orthonormal Wavelet code, and P could consist of m selected rows from the discrete cosine transform (DCT) matrix. An MVM with Φ costs $O(n)$, an MVM with P is $O(n\,log\,n)$, both much faster than a naive $O(nm)$ MVM for large m. Experiments with this setup are in preparation.

6.1 Matrix-free Updates

We already noted that storing X explicitly should not be necessary, if we have an efficient method to compute MVMs with X and X^T. For example, X could be of special structure, or it could be sparse (most entries exactly zero). Suppose that $m \ll n$ with very large n. In this case, the degenerate representation of $Q(a)$ is used, requiring $O(m^2)$ storage. Each EP update requires $O(m^2)$ and the extraction of a column of X, in other words $X\delta_i$, a single MVM (see Section 3.3). It is necessary to refresh the representation now and then, which requires the computation of $X\Pi^{-1}X^T$ for arbitrary

positive π_i. This can be reduced to computing $X\Pi^{-1}X^T\delta_i$, $i = 1,\ldots,m$, corresponding to m MVMs with X and X^T each.

6.2 No Representation At All

It is in general not feasible to compute all required marginals on demand, just storing π, b, and the data. A representation is used in order to do this feasibly, using the fact that each update leads to a rank one modification only. Time is traded for memory, as explained in Section 3.2.

However, suppose that MVMs with X and X^T can be done very efficiently. For an update at site i, we require $Q(a_i) = N(h_i, \sigma^2\rho_i)$. Recall that $\Sigma^{-1} = X^TX + \Pi$ and $h = \Sigma(b^{(0)} + b)$. The quadratic criterion

$$q(v) := \delta_i^T v - (1/2)v^T(X^TX + \Pi)v$$

can be minimised using the linear conjugate gradients (LCG) algorithm (Saad, 1996), requiring a MVM with $X^TX + \Pi$ per iteration, thus MVMs with X, X^T, and $O(n)$. At the minimum, we have $v_* = \Sigma\delta_i$ and $q(v_*) = \rho_i/2$, whence $h_i = v_*^T(b^{(0)} + b)$.

We can also start from the degenerate representation and formulate the marginal computation as quadratic minimisation over vectors of size m, where the system matrix is $I + X\Pi^{-1}X^T$. However, both variants have the same cost of two MVMs per iteration, and their convergence behaviour should be similar.

This method has the advantage of not requiring any representation at all. Apart from the architecture for computing X and X^T MVMs, we need $O(n)$ memory only. However, it is useful only if LCG converges to satisfying accuracy rapidly (after many fewer than n iterations), and if single MVMs can be done much faster than $O(nm)$. Another drawback is that the marginal computations are approximate only, and the error may well depend on the current π. Therefore, it is maybe most sensible to combine it with a way of reducing the number of EP updates required, as is discussed just below.

6.3 Keeping Marginal Moments Up-to-date

The suggestions so far try to speed up single EP update computations. However, if n is very large, a major problem is that we cannot update all n sites in a sweep. For example, in the context of experimental design, it is not affordable to update each site after each new data point inclusion. Updates have to be done selectively on a subset. In this section, we indicate how this can be done. See Seeger and Nickisch (2008) for a demonstration in practice.

Given the marginal $Q(a_i)$, an EP update at i is $O(1)$, so its effect on $Q(a_i)$ can be measured cheaply. The costly part (in the formulation used so far) is to extract the marginal from the representation, and to update the latter. It is reasonable to assume that a small impact of an EP update on the marginal $Q(a_i)$ implies that the whole posterior Q changes little, so site i need not be updated at the moment.

In order to direct EP updates towards sites with maximum marginal impact, it is necessary to keep *all* marginals $Q(a_i)$ up-to-date at all times. In other words, $Q(a_i)$ must be computable in $O(1)$ from the representation, for any i. With the representations of Section 3.2, this costs $O((\min\{n,m\})^2)$. We concentrate on the degenerate representation, which is more important in the large-scale context (the non-degenerate case is also simpler). Let $V := X^TL^{-T}$, and define

$e_1 = \mathrm{diag} VV^T$, $e_2 = V\gamma$. Given the latter vectors, each marginal can be computed in $O(1)$:

$$\rho = \Pi^{-1}(1 - \Pi^{-1} e_1), \quad h = \Pi^{-1}(b^{(0)} + b - e_2).$$

We show how e_1, e_2 can be updated along with the representation. Recall Section 3.2, $x = X_{\cdot,i}$. We can compute Δ_1, Δ_2 and $Q'(a_i)$ in $O(1)$, without knowing $v = L^{-1}x = (V_{i,\cdot})^T$. If $|\Delta_2|$ or $\mathrm{D}[Q'(a_i) \,\|\, Q(a_i)]$ is too small, the update is rejected. Otherwise, we compute v and $w := L^{-T}v$, $r := X^T w$. First, $\tilde{\gamma} = \gamma + \Delta_1 v$. The Woodbury formula gives

$$\left(L'(L')^T\right)^{-1} = \left(LL^T\right)^{-1} - (\Delta_2^{-1} + e_{1,i})^{-1} ww^T,$$

so that $e_1' = e_1 - (\Delta_2^{-1} + e_{1,i})^{-1} r \circ r$, and

$$e_2' = V'(L')^{-1}(L\tilde{\gamma}) = e_2 + (\Delta_1 - (\Delta_2^{-1} + e_{1,i})^{-1} v^T \tilde{\gamma}) r.$$

Now, $v^T\tilde{\gamma} = e_{2,i} + \Delta_1 \|v\|^2 = e_{2,i} + \Delta_1 e_{1,i}$, so that

$$e_2' = e_2 + \frac{\Delta_1 - \Delta_2 e_{2,i}}{1 + \Delta_2 e_{1,i}} r.$$

Finally, L', γ' are obtained from $L, \tilde{\gamma}$ by a rank one Cholesky update as before. The cost is increased by the computation of w and r, the latter requires a single MVM with X^T.

The representation has to be recomputed now and then. Here, the computation of e_1 is most challenging, but can be reduced to doing m MVMs with X^T (with the columns of L^{-T}, obtained by back-substitutions).

In an experimental design context, the representation has to be updated once new data points (x_*, u_*) are included, as discussed in Section 3.4. Using the notation there, V is transformed to V' by appending the column $v := l_*^{-1}(x_* - X^T L^{-T} l)$, where (l^T, l_*) is the new row of L'. Therefore, $e_1' = e_1 + v \circ v$. Moreover, $\gamma' = ((\gamma + u_* l)^T, g_*)^T$ for a scalar g_*, so that

$$e_2' = V(\gamma + u_* l) + g_* v = e_2 + u_* x_* + (g_* - u_* l_*) v.$$

The computation of v requires one MVM with X^T.

Once every marginal is available in $O(1)$ at all times, we can actively select which one to update next. For a set of candidates i, we compute score values S_i, selecting site $\mathrm{argmax}_i S_i$ for the next update. A possible score is $S_i = \mathrm{D}[Q'(a_i) \,\|\, Q(a_i)]$. Scoring all sites for each update is $O(n)$, thus prohibitive, so a set of scoring candidates J should be maintained and evolved. A simple rule, which has been used in the context of sparse Gaussian process methods (Lawrence et al., 2003), works as follows. Before each update, all sites in J are scored. The winner is chosen for the update, and is removed from J, along with a fraction (say, $1/2$) of the worst-scored ones. J is then filled up again by drawing at random from $\{1, \ldots, n\} \setminus J$.

Finally, we note that it is possible in principle to maintain e_1, e_2, therefore the marginal moments, without storing a representation of size $O(m^2)$ at all. For example, the update after a change of π_i, b_i is in terms of $r = X^T L^{-T} v = VV^T \delta_i$. Since $VV^T = \Pi - \Pi\Sigma\Pi$, we have that $r = \pi_i(\delta_i - \Pi\Sigma\delta_i)$. We have shown above how to approximate $\Sigma\delta_i$ by the LCG algorithm. Equivalently, $VV^T = X^T(I + X\Pi^{-1}X^T)^{-1}X$, so we can also compute r by LCG on a system of size m. In principle, such a representation-free method can be used to address problems with large m. However, when working without a representation, we have no efficient possibility anymore to "refresh"

e_1, e_2 now and then. Moreover, using LCG is an additional source of approximation errors. The danger is that (e_1, e_2) and (π, b) drift away from the relation that binds them with exact computations. Experiments assessing the usefulness of such a representation-free treatment in comparison to a $O(m^2)$ representation are in preparation.

7. Experiments

In this section, we present experiments for gene regulatory network identification, for sparse coding of natural images, and for compressive sensing.

7.1 Regulatory Network Identification

Our application of experimental design to gene network identification, using the sparse linear model, has been described in Section 2.2. The material presented here is extracted from Steinke et al. (2007), where all details omitted here can be found. The experiments were done by Florian Steinke. Note that `Matlab` code is available[23] for scientific use. The results given here can be reproduced with this code.

In order to evaluate our method, we simulate the whole network identification process. First, we generate a biologically inspired ground-truth network together with parameters for a numerical simulator of nonlinear dynamics, respecting the network. We feed our method with a number of candidate perturbations $\{u_*\}$, among which it can choose the experiments to be done. If (say) u_* is chosen, a corresponding x_* is drawn from the simulator, and (u_*, x_*) is included into the posterior $Q(A)$ as new observation. We score the predictions from the current posterior against the true network after each inclusion.

Our generator samples networks with a scale-free edge distribution, using $n = 50$ nodes with in-degrees (excluding self-edges) in $\{0, \ldots, 6\}$. An edge is activating with probability $1/2$, inhibitory otherwise. For a given network structure, we sample plausible interaction dynamics, using noisy Hill-type kinetics inspired by the model of Kholodenko et al. (2002). Here, systems without a stable fixed point are rejected.

The disturbance candidates u_* were restricted to have a small number r of non-zero entries, since a tightly controlled excitation or inhibition for many genes at the same time is unreasonably expensive in practice. All non-zero elements have the same size, but a random sign, so that all u_* have the same norm. We use a pool of 200 randomly generated candidates in general.

All results are averaged over 100 runs with independently drawn networks and systems. In the comparative plots presented below, the different methods all run on the same data.

Our evaluation score measures the quality of the ranking of candidate edges, computed from the posterior according to the probabilities $Q(\{|a_{ij}| > 0.1\})$. We modify a standard ROC curve (true positive rate (TPR) as function of false positive rate (FPR)) by computing the area under the ROC curve (AUC) only up to a number of false positives equal to the number of edges in the true network. Namely, since true networks are sparse, there are many more non-edges than edges, and only very small FPRs are acceptable at all. We denote our score as *iAUC*, it is normalised to lie in $[0, 1]$. For $n = 50$, the trivial method which outputs a random permutation as ranking, has expected iAUC of 0.02. Furthermore, on average about 25% of the true edges are "undetectable"

23. See `www.kyb.tuebingen.mpg.de/sparselinearmodel/`. The code is joint work with Florian Steinke and Koji Tsuda. If you use it as part of a scientific publication, please cite Steinke et al. (2007) (details are on the web site).

after linearisation: their entries a_{ij} are very close to zero, so they do not contribute to the dynamics within the linearisation region. Such edges were excluded from the computation of iAUC.

Our method comes with two hyperparameters: the noise variance σ^2, and the scale τ of the Laplace prior. Given sufficient data, they could be estimated by the method described in Section 5, but this is hard to do in an experimental design setting, where we start with very few observations.[24] It is reasonable to assume that a good value for σ^2 does not change too much between networks with similar biological attributes, so that we can transfer it from a system whose dynamics are known, or for which sufficiently many observations are already available. This transfer was simulated in our experiments by generating 50 networks with data as mentioned above, then estimating σ^2 from the size of the ε residuals. The prior parameter τ was set by a simple heuristic described in Steinke et al. (2007).

We used fractional EP with $\eta = 1/2$. Standard EP (i.e., $\eta = 1$) does not converge for the majority of the inference tasks required. This problem is discussed in Section 3.3.1.

In Figure 2, we present reconstruction curves for our method versus competing techniques, which lack novelties of our approach (experimental design, Laplace prior). Very clearly, experimental; design helps to save on costly experiments. The effect is more pronounced for the Laplace than for the Gaussian prior. The former is a better prior for the task, and it is usually observed that improvements of designed over random experiments scale with the appropriateness of the model. In this case, the iAUC level 0.9 is attained after 36 experiments with designed disturbances, yet only after 50 measurements with randomly chosen ones, thus saving 30% of the experiments. In fact, our results indicate that experimental design only realises its full potential together with the non-Gaussian sparsity prior (see also Section 2.1).

In general, the model with Laplace prior does significantly better than with a Gaussian one. Of course, τ for the Laplace and the variance for the Gaussian prior were selected independently, specific to the prior. The difference is most pronounced at times when significantly less than n experiments have been done and the linear system (3) is strongly underdetermined. This confirms our arguments in favour of the Laplace prior (see Section 2.1).

The under-performance of the most direct variant LD of our method, up to about $n/2$ observations, is not yet completely understood. However, it has been repeatedly observed that aggressive experimental design based on very little knowledge can perform worse than random data sampling, if the model does not perfectly reflect the truth. On the other hand, it is important to note that LD recovers completely from the initial under-performance, and from $m = 25$ onwards significantly outperforms the random variant LR, so the initial design choices are not just plain wrong. We also tested a hybrid strategy LM of starting with random, then switching to designed experiments. In this particular application, starting from no knowledge about the network, an initial random exploration seems to lead to most useful results early on, while not hurting a subsequent sequential design.

7.2 Sparse Coding of Natural Images

The application of the sparse linear model to image coding (Olshausen and Field, 1997; Lewicki and Olshausen, 1999) is motivated in Section 2.3. Here, we present results of a study along the lines of work reported by Olshausen and Field (1997).[25] As is argued at length in Section 2.3, our goal here

24. One may be able to correct initial estimates of σ^2, as more observations are made, and a method for doing so is subject to future work.

25. Data and code used there was obtained from `http://redwood.berkeley.edu/bruno/sparsenet/`.

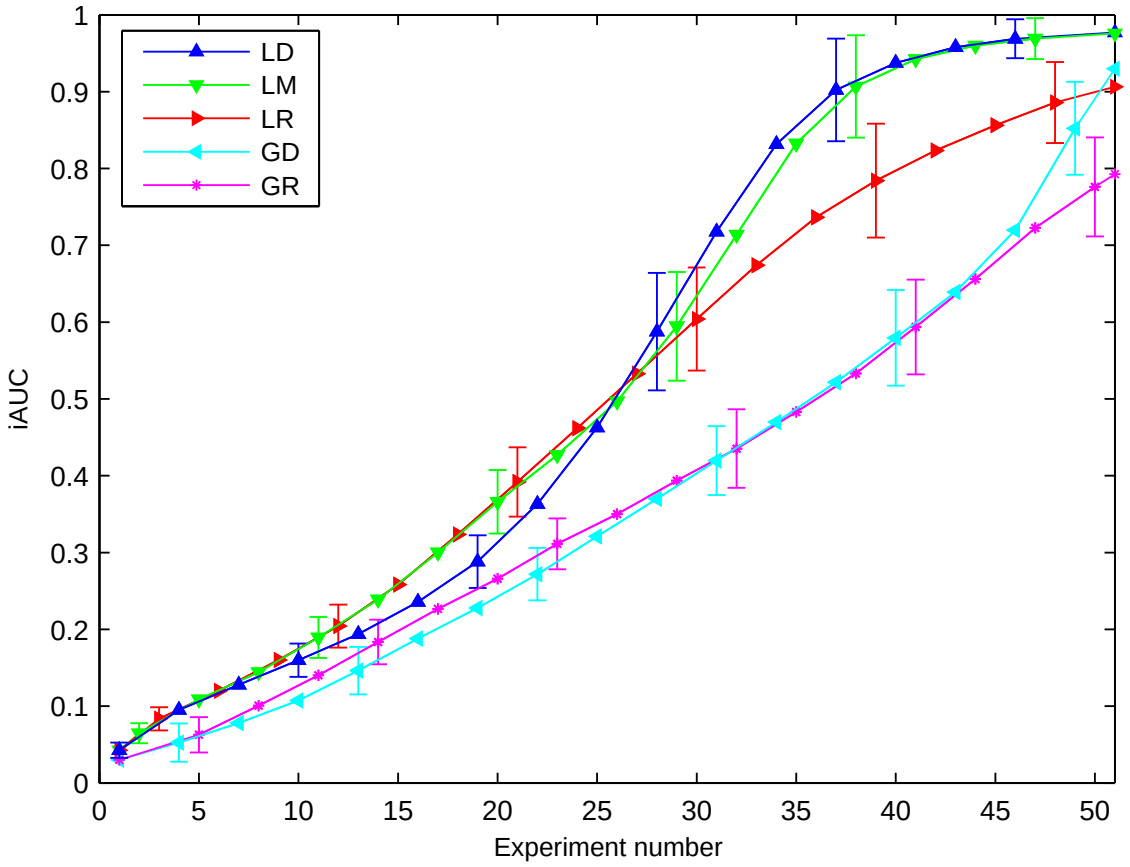

Figure 2: Reconstruction curves for experiments (gene expression changes of 1%, SNR 100, $r = 3$ non-zeros per u). **LD**: Laplace prior, experimental design. **LR**: Laplace prior, random experiments. **GD**: Gaussian prior, experimental design. **GR**: Gaussian prior, random experiments. **LM**: Laplace prior, mixed selections (first 20 random, then designed). Error bars show one standard deviation over runs. All visually discernible differences in mean curves of different methods are significant under the t-test at level 1%.

is not to compare a range of models to find out which can code images better or learn codes more efficiently, but rather to test the hypothesis put forward in Lewicki and Olshausen (1999), which does not call for such a comparison. We extracted two data sets of $r = 50000$ image patches of size 12×12 by subsampling the 10 whitened natural scenes, using their Matlab code. One is for training, the other for evaluation.[26] We allow for a twice overcomplete basis, therefore $m = 144$, $n = 288$. We also drew a random subsample of size 1000 from the test set, which was used in order to produce curves over many codes. Recall that a code in our model is given by the matrix X, whose columns are referred to as codebook vectors or filters.

26. The sets are not guaranteed to be completely distinct, although the extraction of the same patch during the different sampling runs is unlikely.

Before we describe the results, we should stress that the main aim at this point is to demonstrate the efficiency and usefulness of our method on a learning problem of large scale. 41473 hyperparameters are learned here on a set of 50000 cases. Each update step requires approximate inference on 100 models, each of which comes with 288 latent parameters. A more careful study will explore the implications of our findings to neuroscience (early vision) and image coding. This will entail a more refined learning schedule for our method, while our choices here are ad hoc and did not receive much tuning. Moreover, we base our evaluation entirely on the EP marginal likelihood approximation on a test set. An independent MCMC evaluation of this criterion, using for example annealed importance sampling together with hybrid Monte Carlo (which seems commonly accepted, but is very expensive to run), is clearly needed in order to draw any scientific conclusions (which we refrain from doing here). Such a study is subject to future work, and is not in the scope of this paper.

As noted above, the approach of Olshausen and Field (1997) is to approximate inference by maximum a posteriori (MAP): $P(a_j|u_j,X) \approx \delta_{\hat{a}_j}(a_j)$, where $\hat{a}_j$ is the posterior mode. Since the log posterior is concave, this mode is unique and can be found efficiently. In order to learn the code X, they propose to impute their estimates in order to obtain a complete data set $\{(u_j,\hat{a}_j)\}$, then to do maximum likelihood training. Since the estimate $\hat{a}_j$ depends on the current X, this is an iterative process. Their method will be called OF in the sequel. In contrast to this, we follow the hypothesis of Lewicki and Olshausen (1999) and learn X by maximising the EP approximation to the log marginal likelihood $\log P(D|X)$ (see Section 5). While Lewicki and Olshausen (1999) argue that the method of Olshausen and Field (1997) can be seen as optimising a (different) surrogate to $\log P(D|X)$ as well, ours is a much better approximation in general.[27]. Our method will be called EP here.

We had to modify their code in a minimal way, in order for it to run automatically on a given fixed training set. Our changes are detailed in Appendix E. Just as with our own method, we did not attempt to refine parameters for their code here.

The code of Olshausen and Field (1997) performs stochastic gradient descent on batches of size $|B| = 100$. We use a similar approach, which works as follows. The criterion $-\log P(D|X)$ is a sum of independent parts, one for each image. Let $\phi := -\sum_{j\in B} L_j$ be the EP approximation to this criterion, evaluated over a batch of size $|B|$ (here, L_j is the EP log marginal likelihood approximation on image j). The update rule for X is

$$X' = X - D', \quad D' = 0.85D + \xi X X^T \nabla_X \phi,$$

where $\xi > 0$ is the learning rate. The pre-multiplication of the gradient by XX^T is advantageous for this application, as has been argued in the context of "natural gradient" learning. As opposed to Olshausen and Field (1997) and Lewicki and Olshausen (1999), we adjust the noise variance σ^2 in the same way by minimising ϕ. If $l := \log\sigma^2$, a simple update rule is

$$l' = l - d', \quad d' = 0.85d + \xi_l \nabla_l \phi,$$

where $\xi_l > 0$ is a learning rate different from ξ. The learning rates are decreased in a reasonably slow way,

$$\xi(t) = \frac{A}{B+t}, \quad \xi_l(t) = \frac{A_l}{B_l+t},$$

27. MCMC experiments to strengthen this claim are subject to future work.

where t is the number of updates so far, and A, B, A_l, and B_l are free parameters (Bottou, 1998).

We can compare our update rule of X with the one used in Olshausen and Field (1997). It is easy to see that the gradient of the exact log marginal likelihood is

$$\nabla_X \log P(u_j|X) = \sigma^{-2} \mathrm{E}_{P(a_j|u_j,X)}[e_j a_j^T] = \sigma^{-2} \left((u - X\,\mathrm{E}[a_j])\mathrm{E}[a_j]^T - X\,\mathrm{Cov}[a_j] \right),$$

where $e_j := u_j - X a_j$. If OF is seen as optimising an approximation thereof, then the expectation over $P(a_j|u_j,X)$ is replaced by plugging in the mode $\hat{a}_j$ (which is what we mean by "imputation" above). In other words, the posterior mean $\mathrm{E}[a_j]$ is replaced with the mode, and the second term depending on the posterior covariance $\mathrm{Cov}[a_j]$ is neglected altogether. Since the Laplace sparsity prior leads to a posterior which is significantly skewed (towards coordinate axes, see Figure 1), mean and mode tend to be quite different.[28] In EP, the posterior expectations are replaced by $\mathrm{E}_Q[\cdot]$, where $Q = N(h, \sigma^2\Sigma)$ is the EP posterior approximation (see Appendix C). In fact, running OF precisely with the learning rule just stated does not work well in practice, and the neglectance of posterior uncertainty in the learning rule is put forward as a reason for this in Lewicki and Olshausen (1999). Olshausen and Field (1997) propose a heuristic renormalisation of the columns of X towards some "desired variance" as remedy, and this seems an important feature in their code. This heuristic comes with a number of parameters, which are fixed in their code to some values presumably optimised for their data by hand. In contrast, EP comes with τ, σ^2 only, and the latter can be adjusted automatically as well,[29] as is demonstrated here. Note that Lewicki and Olshausen (1999) suggest to approximate $\mathrm{Cov}[a_j]$ by the Laplace method in order to improve on the OF learning rule. However, as noted in Section 3, this method is not well-defined in case of the sparse linear model. Code implementing the proposal of Lewicki and Olshausen (1999) is not publicly available.

We can draw an analogy between the difference of learning X in OF and EP to current practices in speech recognition (Rabiner and Juang, 2003). Given a trained system, the recognition (or decoding) is done by searching for the most likely sequence, in what is called Viterbi decoding. However, training the system should be done by expectation Maximisation (EM), where the latent sequence is integrated out using inference. This is about what EP does here, with the difference that inference in hidden Markov models used for speech is analytically tractable, but has to be approximated here (by EP). However, since EM training is still computationally demanding, most speech recognition systems use Viterbi training today, where just as in OF the most likely (MAP) sequence is imputed instead of doing inference. While EM training is known to produce better recognisers on the same data, MAP training is still preferred for reasons of computational efficiency.

Our setup is as follows. We ran all methods on the same training data set, starting from the same initial code (drawn at random). For OF, learning rate and renormalisation heuristic parameters were left unchanged in their code. We used the values $0.1, 0.2, 0.4, 0.6962$ for τ, and $0.006, 0.01$ for σ^2. The values $\tau = 0.6962$, $\sigma^2 = 0.01$ come from the OF code, while $\sigma^2 = 0.006$ is closer to values ultimately preferred by the EP runs. All methods were run for 10000 batch updates, thus 20 sweeps over all images (in random ordering, different for each sweep). OF was run[30] separately for each of the eight (τ, σ^2) variants. On the other hand, for the EP runs, σ^2 was adjusted along with X as described above, and only τ was provided (the initial value for σ^2 was 0.002). The following

28. As discussed in Section 2.1, the mode $\hat{a}_j$ has many components which are exactly zero, which does not hold for the mean.
29. We could adjust τ in the same way with EP, but this is not done here. As noted in Section 3.5, the optimisation of σ^2 should behave better.
30. We also ran OF with $\tau = 0.05$, which gave bad results not reported here.

	$\tau = 0.1$		$\tau = 0.2$		$\tau = 0.4$		$\tau = 0.69$	
σ^2	OF	EP	OF	EP	OF	EP	OF	EP
0.006	-27.09	-80.04	-68.17	-80.04	-35.91	-80.05	92.73	-80.04
0.01	-2.329	-63.37	-53.53	-63.47	-43.69	-63.63	60.44	-63.80

Table 1: EP negative log marginal likelihood (EP average coding cost) per image, evaluated on the full test set (50000 cases), for different methods after 10000 batch updates of learning X. OF: Olshausen/Field; EP: our method.

learning rate schedule parameters[31] were used: $A = 0.79$, $B = 0.79 \cdot 10^3$, $A_l = 0.79 \cdot 10^{-5}$, $B_l = B$. This means that ξ decreases from 10^{-3} to $7.32 \cdot 10^{-5}$, and ξ_l from 10^{-8} to $7.32 \cdot 10^{-10}$.

We compare methods in general by evaluating the EP negative log marginal likelihood approximation on the test set, normalised by the number of images. This is equivalent to the EP approximation of the average coding cost per image (Lewicki and Olshausen, 1999) (smaller is better). For all but the final codes (after 10000 updates), we do this evaluation on the subset of size 1000. In order to evaluate test scores or to learn X (EP variants only), we need to perform EP inference on each image separately. To this end, we intended to use standard EP initially ($\eta = 1$, see Section 3.3.1), but ran into severe numerical problems on a significant number of images,[32] as described in Section 3.3.1. This led us to use fractional EP with $\eta = 0.9$ instead, which is the basis for all learning and test score evaluation results presented in this section.

learning curves along 10000 batch updates are shown in Figure 3 (using the test subset), and final EP negative log marginal likelihoods per image on the full test are given in Table 1. To recapitulate, the figures show average coding costs per image under the codes learned by the different methods, where τ and σ^2 are fixed for the evaluation. While OF was provided with τ, σ^2, EP only received τ during learning and had to adjust σ^2 alongside the code matrix X. We see from Figure 3 (upper left) that for the learning rate schedule used here, EP grows σ^2 smoothly from 0.002 to about 0.005.

Note that in the Bayesian viewpoint of image coding, all hyperparameters of a model work together in order to represent a data distribution (of image patches u) well, that is the code X, but also τ and the noise variance σ^2. In other words, code and noise variance are dependent. The EP runs settle at around $\sigma^2 = 0.005$, so the codes X found by them do better at $\sigma^2 = 0.006$ than at $\sigma^2 = 0.01$. The results for EP seem to not much depend on τ, but the situation is quite different for OF. At $\tau = 0.2$, the OF codes do well in comparison to the EP ones, and the lower $\sigma^2 = 0.006$ is preferred as well. For $\tau = 0.1$ or $\tau = 0.4$, they do significantly worse, and the preferred value is

31. These values were chosen after few initial runs, but not optimised over. Only for $\xi_l(0)$ did we compare runs, looking at learning curves on the training set. For $\xi_l(0) = 10^{-9}$, σ^2 hardly changed at all, while for $\xi_l(0) = 10^{-7}$, σ^2 increased sharply to above 0.02, then descended slowly towards 0.01.

32. None of these problems happened with fractional EP, $\eta = 0.9$. However, there is a pattern to these failures, indicating that further analysis would be valuable. In general, during learning X, EP convergence was harder to attain when the code was already optimised, with structural features emerging in the filters. While X could still be learned with $\tau = 1$, the test set log marginal likelihood evaluations for these codes could not be computed for many patches (using EP with $\eta = 1$). We evaluated these scores using EP with $\eta = 0.9$, finding very similar results (not shown here) than with the codes learned using $\eta = 0.9$. The reason for not simply abandoning standard EP for more robust fractional variants in general is based on arguments concerning alpha-divergences (Minka, 2004) (no hard theory is available to settle this issue, to our knowledge), apart from the somewhat more appealing motivation that can be given for standard EP (Opper and Winther, 2000).

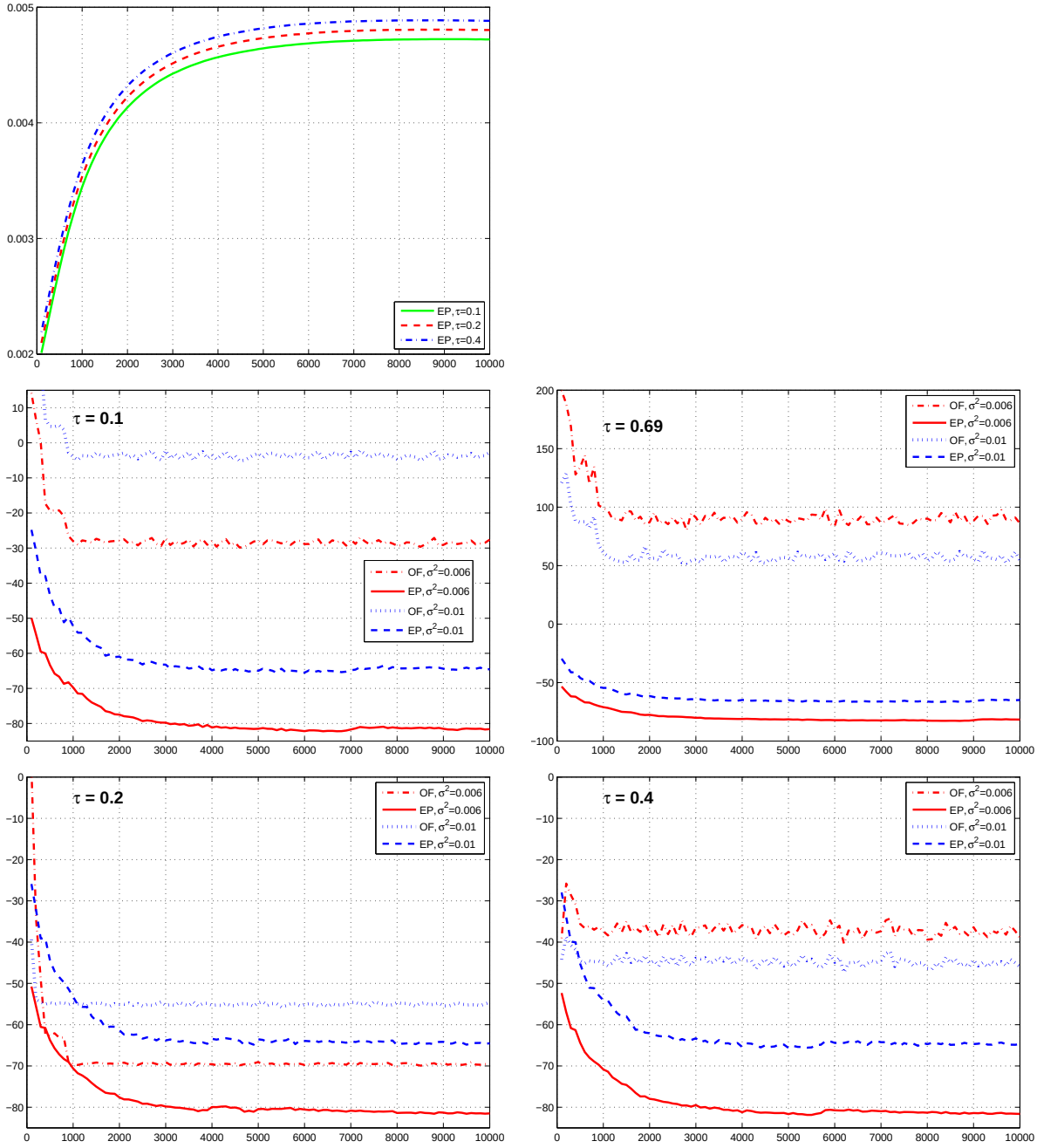

Figure 3: learning curves along 10000 batch updates. Upper left: Noise variance σ^2 for EP (different prior scales τ). Others: EP $-\log P(D)/r$ on test subset ($r = 1000$); $\tau = 0.1$ (middle left), $\tau = 0.6962$ (middle right), $\tau = 0.2$ (lower left), $\tau = 0.4$ (lower right).

$\sigma^2 = 0.01$ for $\tau = 0.4$. Finally, poor results[33] are obtained by OF with $\tau = 0.6962$, as well as with

33. We re-ran this case several times, in the way described in Appendix E, always obtaining the same poor results.

$\tau = 0.05$ (not shown here). The learning curve behaviour of the EP runs is much smoother than for OF, suggesting that the former optimisation problem is better behaved.

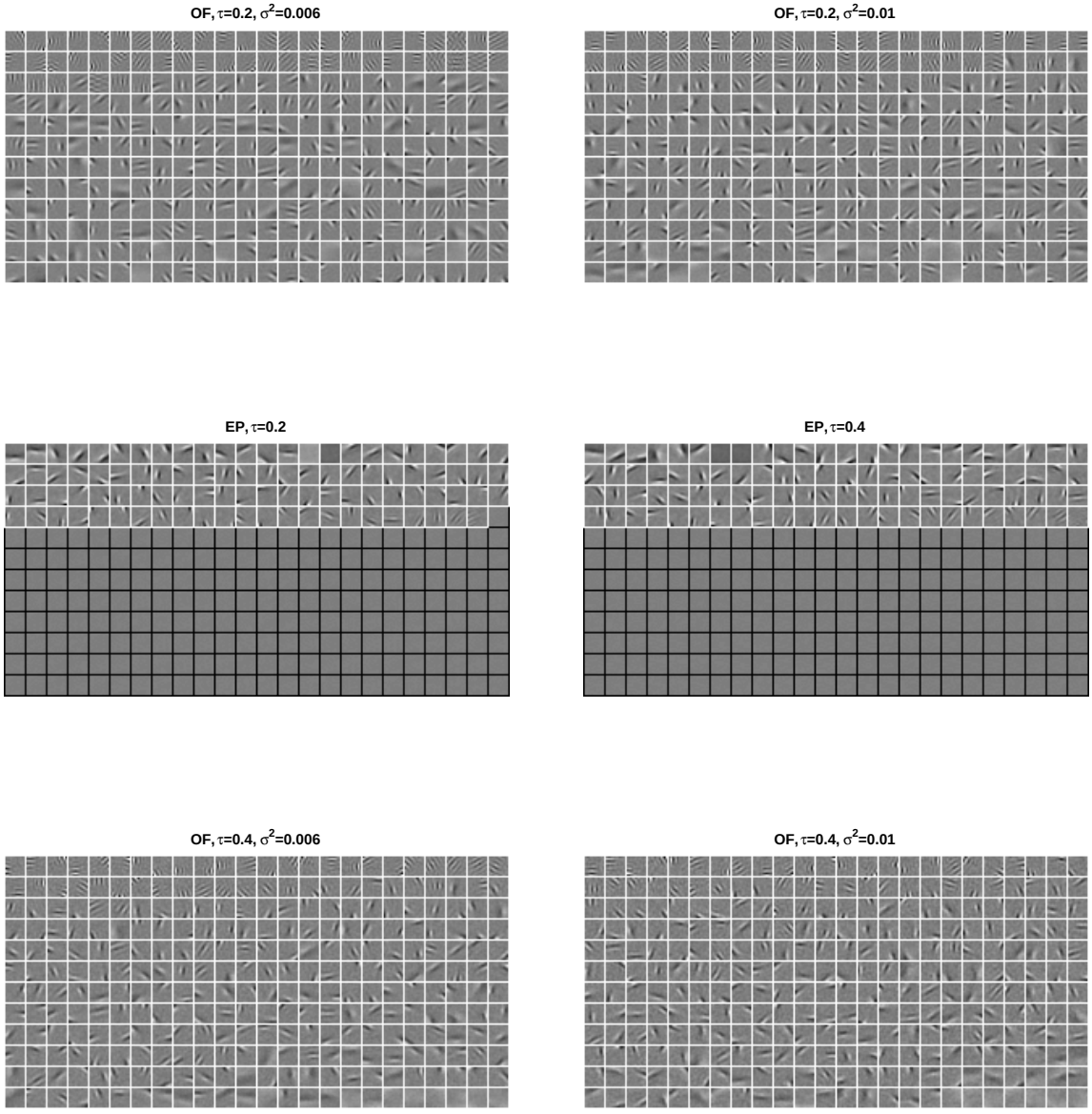

Figure 4: Final codes (after 10000 batch updates). Filters are ordered by descending $\|X_{\cdot,i}\|$, row-major ordering. Filters with $\|X_{\cdot,i}\| > (3\%) \max_j \|X_{\cdot,j}\|$ have white frame, black otherwise.

	$\tau = 0.1$		$\tau = 0.2$		$\tau = 0.4$		$\tau = 0.69$	
σ^2	EP[96]	EP[288]	EP[95]	EP[288]	EP[96]	EP[288]	EP[94]	EP[288]
0.006	-81.41	-80.04	-81.41	-80.04	-81.41	-80.05	-81.41	-80.04
0.01	-64.74	-63.37	-64.84	-63.47	-65.18	-63.63	-65.18	-63.80

Table 2: EP negative log marginal likelihood (EP average coding cost) per image, evaluated on the full test set (50000 cases), where X has been learned by EP. EP[288]: All X columns (copied from Table 1); EP[94–96]: Only X columns of significant size.

The final codes for different setups are given in Figure 4. The most distinctive difference between EP and OF codes is that for the latter, the filters do not differ much in size,[34] while there is a clear size signature in the codes found by EP: about 96 filters have significant sizes, while the remaining ones are about two orders of magnitude smaller. In the panels of Figure 4, filters with $\|X_{\cdot,i}\|$ larger than three percent of the maximum value are surrounded by white frames. For EP, these are 94–96 of 288 columns of X. In Table 2, we show EP average coding costs for the full test set, given that only the filters of significant size are used. These are even slightly lower than for the respective models using all columns of X.

We see that for the given task, codes attaining the lowest average cost are in fact *undercomplete*. A Bayesian method (such as EP here) removes unnecessary dimensions by default, through what has been called automatic relevance determination (see also Section 8.1). This does not happen for the OF method, which is not Bayesian and ignores covariances when learning X. We also note that in the codes found by OF, the filters of largest size are non-localised gratings, while all filters of significant size found by EP are localised and oriented. Both the smaller number of filters required and the strict localisation properties can be explained by noting that each image patch is explained probabilistically in EP, following Lewicki and Olshausen (1999), while in OF, this has to be done using a deterministic sparse encoding. In an update of X, each image only affects a small number of filters. It is then not too surprising that additional non-localised filters emerge in OF. If the hypothesis of Lewicki and Olshausen (1999) is taken for granted, these filters should be interpreted as artifacts of its improper implementation.

Note that the OF method runs much faster than EP. Finding $\hat{a}_j$ is a quadratic program (Tibshirani, 1996), which can be solved efficiently. Our EP code for the experiments here is "naive", in that all sites are visited in random ordering, no further efforts (such as the ones described in Section 6.3) are done. However, the arguments in Olshausen and Field (1997) and Lewicki and Olshausen (1999) do not call for a method which can be run very efficiently on a digital computer. A model is suggested which, in simple terms such as independence, linearity, and sparsity, could account for the formation of early visual neuron's receptive fields. The hypothesis of Lewicki and Olshausen (1999) is equivalent to a Bayesian perspective, where inference is a core requirement for improving the code, in much the same way as in EM for speech recognition, or graphical model learning in general. For both OF and EP, filters of significant size are localised, oriented gratings. However, our EP method more accurately implements the hypothesis of Lewicki and Olshausen (1999) than the algorithm of Olshausen and Field (1997), and leads to *qualitatively different* find-

34. Their renormalisation heuristic keeps them at similar, yet not at equal sizes.

ings: the data calls for a significantly undercomplete, therefore rather compact code, a fact that is not picked up by the OF method at all (Berkes et al., 2008, report similar findings with an approximate Bayesian method). Moreover, OF uses a significant number of non-localised filters, which are not present among the vectors of significant size found by EP. A more careful study based on our framework will shed light on what relevant properties in the codes can be explained by the probabilistic hypothesis of Lewicki and Olshausen (1999), versus which findings should rather be attributed to their particular computational method (maximum a posteriori, winner-takes-all X updates, variance renormalisation heuristic, etc.).

Apart from learning codes with EP, we can also use the log marginal likelihood approximation of EP in order to compare codes obtained by other methods. In Bayesian terms, such a comparison is done by computing Bayes factors, which is comparable to hypothesis testing. Moreover, for a fixed code X and data $\{u_j\}$, the noise variance σ^2 can be optimised by EP, in what is suggested to be a robust process in Section 3.5.

7.3 Compressive Sensing

In this section, we present results for a compressive sensing toy example. The motivation behind this application was given in Section 2.4. Results from a larger set of experiments, including some large-scale applications (see Section 6), are given in a later paper (Seeger and Nickisch, 2008). The experiments have been done by Hannes Nickisch.

In our toy experiment, the signal $y \in \mathbb{R}^n$ is sparse itself, so the coding matrix Φ is the identity. We have $n = 512$. Measurements are taken as $u = Py + \varepsilon$, where P (or X here) is the measurement matrix, and ε is Gaussian noise with standard deviation $\sigma = 0.005$. We compare methods where the measurement projections P are optimised in a sequential row-by-row manner, with methods where P is drawn uniformly at random on the unit hypersphere. In any case, the projections (i.e., rows of P) are constrained to have unit norm. The signal y is created by drawing $k = 20$ non-zero positions at random. The non-zero y_i are drawn at random from $\{-1,+1\}$ (uniform spikes), or according to a density[35] with support $\mathbb{R} \setminus (-0.21, 0.21)$ (non-uniform spikes). Examples for such signals are shown in Figure 5.

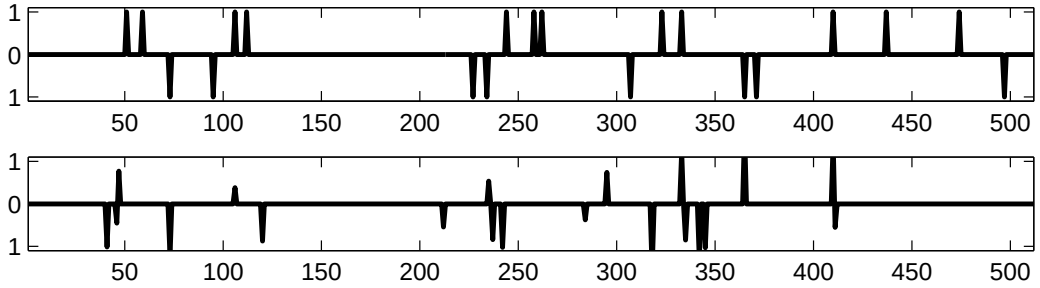

Figure 5: Examples for signal y. Top: uniform spikes. Bottom: non-uniform spikes.

The sequential experimental design of rows of P is a special case of the standard design setup of Section 4. Namely, among all p_j of unit norm, select the one which leads to minimum expected entropy $\mathrm{E}[\mathrm{H}[Q']]$, where Q' is the posterior after inclusion of p_j, and the expectation is w.r.t. $Q(u_j)$,

35. Namely, $y_i = \alpha(r + 0.25\,\mathrm{sgn}\, r)$, $r \sim N(0,1)$, where $\alpha = (5/4 + \sqrt{2/\pi} - 2/\pi)^{-1/2} \approx 0.84$, so that $\mathrm{Var}[y_i] = 1$.

u_j being the new measurement. Note that $\mathrm{H}[Q']$ is in fact independent of u_j, since Q' is Gaussian. Moreover, one can show that this criterion is equivalent to the expected information gain in this case (MacKay, 1991). A simple argument shows that the eigenvector for the largest eigenvalue of Σ solves this problem, where $Q = N(h, \sigma^2\Sigma)$ is the current posterior. This eigenvector can be found by the power method.

We compare the following methods. Our design approach based on EP is called *EP opt*. The method suggested by Ji and Carin (2007) is called *RVM opt* (P designed) or *RVM rand* (P random). They select P in the same way as we do, but making use of their approximate posterior, which they obtain as a variant of sparse Bayesian learning (SBL) (Tipping, 2001) (RVM refers to the most commonly used variant of SBL). The method most frequently used in compressive sensing applications so far is basis pursuit (Chen et al., 1999), where y is estimated by minimising $\|y\|_1 = \sum_i |y_i|$, subject to $Xy = u$. Note that this corresponds to MAP estimation in the sparse linear model if $\sigma^2 \to 0$ (noiseless case). This can be formulated as a linear program. *L1* and *BP* here use two different implementations.[36] For all methods, the first 40 rows of P are drawn at random. If $\hat{y}$ denotes the best prediction of y from the measurements u (the mean of Q for our method and the RVM variants), the error is measured as $\|\hat{y} - y\|/\|y\|$, where $\|\cdot\|$ is the Euclidean norm. Results are shown in Figure 6.

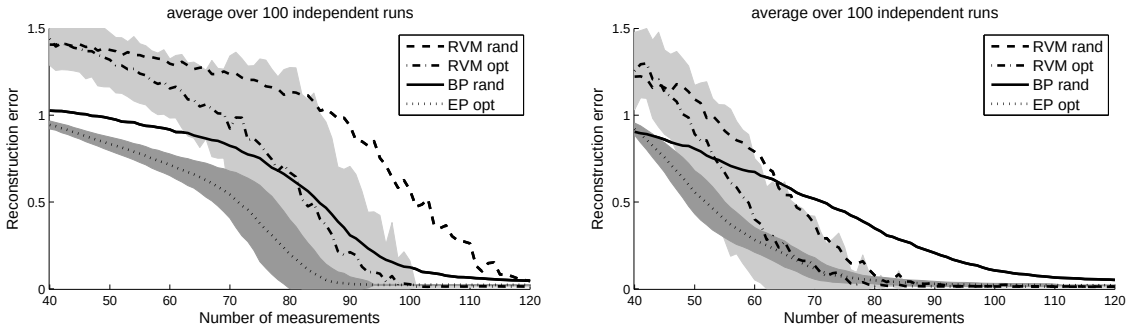

Figure 6: Results for compressive sensing toy example comparison. Left: uniform spikes. Right: non-uniform spikes. Averaged over 100 runs, shown are means and standard deviations (latter only for EP and RVM). See text for details.

Further experiments are required in order to draw definite conclusions, such are in preparation. We note that our EP method outperforms all others, and that random methods in general perform worse than the ones using experimental design. Moreover, our method clearly performs much better than the method of Ji and Carin (2007), while theirs is somewhat faster. Notably, our method also performs more robustly across runs than theirs. Moreover, the methods trying to approximate Bayesian inference in general perform better than basis pursuit on this task. The latter is certainly significantly faster than any of the other methods here, but its suboptimal performance on the same fixed data, and more importantly the lack of an experimental design framework, clearly motivates considering approximate Bayesian inference for compressive sensing as well.

36. *L1* is `l1-magic` from `www.acm.caltech.edu/l1magic/`, and *BP* is from SparseLab `sparselab.stanford.edu/`.

8. Related Approximate Inference Methods

In this section, we put EP for the sparse linear model into perspective by directly comparing it to other proposed methods of approximate inference. We rely on Palmer et al. (2006), who provide a somewhat more general discussion, but EP is not mentioned there. Before we start, we remind the reader that by "approximate inference" method, we mean a technique which delivers a useful approximation of marginal (or even joint) posteriors, and ideally can also be used in order to approximate the marginal likelihood. This is important in the context of the sparse linear model, where many methods rather try to find a maximally sparse solution of the noisy system (1), without addressing the former points. A study, comparing the methods discussed here in terms of approximation quality of marginals and of the marginal likelihood, is subject to future work. Note that none of these methods has been applied to the experimental design problem we address here (to our knowledge), with the exception of SBL (Ji and Carin, 2007).

8.1 Sparse Bayesian Learning

The idea of automatic relevance determination (ARD) has been proposed by Neal (1996). It is a variant of empirical Bayesian marginal likelihood maximisation (see Section 5). In the context of the sparse linear model, only a few components of a are typically relevant for describing the data, all others could be set to zero. ARD works by placing a prior $N(a_i|0,\sigma^2\pi_i^{-1})$ on a_i, where π_i is a scale parameter, then maximising the marginal likelihood $P(u,\pi)$ w.r.t. π. Here, π_i can be given a heavy-tailed hyperprior. The Occam's razor effect embedded in empirical Bayes (MacKay, 1992) leads to π_i becoming large for irrelevant components a_i: a model with few relevant components is simpler than one with many, and if both describe the data well, the former is preferred under ARD.

ARD has been applied to the sparse linear model by Tipping (2001), where the method was called *sparse Bayesian learning* (SBL). The derivation there makes use of *scale mixture* decompositions (Gneiting, 1997; Palmer et al., 2006) for the non-Gaussian prior sites. Namely, many univariate symmetric distributions can be represented in the form $P(a_i) = \mathrm{E}[N(a_i|0,\sigma^2\pi_i^{-1})]$, with some distribution over π_i. Tipping uses Student's t sparsity priors $P(a_i)$, for which π_i has a Gamma distribution. However, a direct comparison with the sparse linear model used here requires Laplace priors.

The Laplace density has the following scale mixture decomposition (Park and Casella, 2005; Gneiting, 1997):

$$\frac{\tilde{\tau}}{2}e^{-\tilde{\tau}|a_i|} = \mathrm{E}[N(a_i|0,\sigma^2\pi_i^{-1})], \quad \pi_i \sim \lambda\pi_i^{-2}e^{-\lambda/\pi_i}\mathrm{I}_{\{\pi_i>0\}} = \mathrm{IG}(1,\lambda),\ \lambda = \frac{\tau^2}{2}. \tag{9}$$

Note that the scale distribution of π_i does not have mean or variance. With $\Pi = \mathrm{diag}\,\pi$, we have

$$P(u,\pi) = \int P^{(0)}(a)N(a|0,\sigma^2\Pi^{-1})\,da|\Pi|^{-2}\lambda^n e^{-\lambda 1^T(\pi^{-1})},$$

which has the same form as in our framework. Here, $b_i = 0$, and the π_i have a different interpretation as scale hyperparameters. The marginal likelihood $P(u)$ is obtained by integrating out π, which cannot be done tractably. Instead, a *maximum a posteriori* (MAP) approximation is done in SBL: we find a maximiser $\hat{\pi}$ of $P(u,\pi)$, then approximate $P(u) \approx P(u,\hat{\pi})$. This is a joint non-convex optimisation problem, so all we can hope for is a local maximum. Faul and Tipping (2002) propose

the simple sequential technique of maximising $P(u,\pi)$ one π_i at a time. This results in the following update rule, as is shown in Appendix D.1:

$$\pi_i' = \frac{\sqrt{9+4\tau^2\beta}-3}{2\beta}, \quad \beta = \rho_i + \sigma^{-2}h_i^2 = \sigma^{-2}\mathrm{E}_Q[a_i^2]. \tag{10}$$

Confusingly, this is in fact not the method used in the experiments of Tipping (2001). This point is clarified at the end of this section.

Comparing SBL to our EP method, we note that the former does not require quadrature, but merely simple analytical updates. The π_i remain positive, and the method is numerically stable. SBL can be implemented using the same representation as ours. While $b = 0$ here, this does not lead to simplifications in representations or updates. In fact, both methods can share much of the same code, they differ only in how $\pi_i \to \pi_i'$ is computed for each site i. The marginal likelihood approximation resulting from SBL is $P(u,\hat{\pi})$. Just as for EP, this is not a bound on $P(u)$ (see Section 5).

Note that a variant of SBL with the Laplace prior has been proposed by Figueiredo (2003). However, they were interested in the MAP solution $\mathrm{argmax}_a P(a|D)$ rather than in an approximation to the posterior, which allowed them to integrate out the π_i by EM. Note also that SBL for the linear model with Student's t prior has been applied to gene network identification by Rogers and Girolami (2005), although they did not consider experimental design.

While a direct comparison is subject to future work, we note that SBL is certainly simpler to implement for the sparse linear model. Some safeguards required to make EP run in a numerically robust way, are not needed with SBL. On the other hand, EP is of course more general, since SBL is limited to non-Gaussian sites with a scale mixture decomposition. For example, non-symmetrical distributions such as classification likelihoods cannot be used.

There is at least the following worrying fact about SBL as approximation to Bayesian inference. We have used the scale mixture decomposition of the Laplace density (9) in terms of π_i, but we could just as well have chosen the one based on $s_i = \pi_i^{-1}$, with an exponential distribution on s_i. Doing so, we obtain an entirely different method, which did much worse than the variant derived here in initial experiments and in fact fails badly as approximation to Bayesian inference, since predictive variances are orders of magnitude too small. Furthermore, this "variant" of SBL converges exceedingly slowly in the s_i, while the method given here runs quite fast. Nevertheless, *both* variants are motivated in the same way: scale mixture decomposition, followed by a MAP approximation. The problem is that the latter, much in contrast to exact Bayesian inference (or, in fact, to expectation propagation), is not invariant to reparameterisations. The one chosen by Tipping (2001) certainly works well, at least in terms of delivering sparse solutions, but with others, SBL can fail badly. This important ambiguity has been noted by Wipf et al. (2004).

Finally, we note that there is some confusion about what exactly constitutes SBL, started in part by somewhat unclear formulations in Tipping (2001). In the paper, a method for finding maximally sparse solutions to the noisy linear system (1) is proposed. While the motivation is clearly Bayesian, the fact that a Student's t sparsity prior is used, is mentioned only in order to explain the favourable results. In fact, the "prior" actually used for π_i is $\propto 1/\pi_i$, resulting in $P(a_i) \propto 1/|a_i|$. Both are not normalisable as distributions. Our interest is in approximate Bayesian inference, with an eye towards experimental design, so we cannot consider such uninformative priors. We take the freedom here to interpret SBL as introducing scale mixture parameters π_i, followed by a MAP approximation w.r.t. π, at the expense of actually not covering the algorithm used in the experiments of Tipping

(2001). Wipf et al. (2004) show that the latter algorithm can in fact be interpreted as an instance of direct site bounding (see Section 8.2), which at first sight has little to do with scale mixtures, but see Palmer et al. (2006).

We would not stress this point if there was little difference in practice between (what we refer to as) SBL and direct site bounding (or other variants of the theme). However, initial comparative experiments with the sparse linear model show very significant differences in approximation quality. All these techniques find local maxima of $P(u|\pi)f(\pi)$. Contrary to what seems to be widely believed among practitioners, the form of f really matters. From our experience, the quality of approximate inference as well as the speed of convergence of sequential optimisation depend strongly on f. Wipf et al. (2007) show that the capability of the method estimating the correct relevant subset also hinges dominantly on the choice of f. Beyond that, the dependence on f of the quality of the covariance estimate, centrally important for experimental design, has not been analysed at all to our knowledge.

8.2 Direct Site Bounding. Variational Mean Field Bayes

A direct approach for obtaining an easily computable lower bound on the log marginal likelihood $\log P(u)$ works by lower-bounding the sites $t_i(a_i)$ by terms of Gaussian form. A powerful way of obtaining global lower bounds of simple form is exploiting convexity (Jaakkola, 1997). We can apply this approach to the sparse linear model with Laplace prior, which results in a method proposed by Girolami (2001). The general idea in the context of non-Gaussian linear models is noted in Palmer et al. (2006).

For the Laplace (2), we have that $\log t_i(a_i) = -\tilde{\tau}\sqrt{a_i^2} + \log(\tilde{\tau}/2)$, which is convex in a_i^2. A global tight lower bound is obtained using Legendre-Fenchel duality (Boyd and Vandenberghe, 2002), resulting in

$$e^{-\tilde{\tau}|a_i|} = \sup_{\pi_i>0} N^U(a_i|0,\sigma^{-2}\pi_i)e^{-(\tau^2/2)\pi_i^{-1}}.$$

We can plug in the r.h.s. for $t_i(a_i)$, then integrate out a in order to obtain a lower bound on $\log P(u)$. The outcome is quite similar to SBL, where t_i is replaced by the same term times $(2\pi)^{-1/2}\tau\pi_i^{-3/2}$. Since the ratio does not depend on a, we have that

$$P(u) \geq P_{Giro}(u;\pi) = (2\pi)^{-n/2}\tau^n|\Pi|^{-3/2}P_{SBL}(u,\pi),$$

Following Appendix D.1, it is clear that the update of π_i, keeping all others fixed, results in a quadratic equation with the positive solution

$$\pi_i' = \frac{\tau}{\sqrt{\beta}}, \quad \beta = \rho_i + \sigma^{-2}h_i^2 = \sigma^{-2}\mathrm{E}_Q[a_i^2].$$

While SBL does not render a bound on $\log P(u)$, Girolami's method does so by construction. Note that SBL and direct site bounding lead to quite similar replacements for t_i, if applied to the linear model with Laplace prior. The same is true if a Student's t prior is used, as has been observed by Wipf et al. (2004). Somewhat ironically, the modification in the latter case is precisely the result of the "uninformative limit" taken in Tipping (2001), which also seems to work best in practice. This point is discussed at the end of Section 8.1. Palmer et al. (2006) give the precise relationship between SBL and direct site bounding (called "integral case" and "convex case" there), showing that if t_i admits a scale mixture decomposition, it can also be bounded via Legendre duality.

Note that for the same (β, τ), π'_i is smaller for the SBL update than for the direct site bounding one. Namely,

$$\pi'_{SBL,i} = \pi'_{Giro,i} \times \left(\sqrt{1+\alpha} - \sqrt{\alpha}\right), \quad \alpha = 9/(4\tau^2\beta).$$

The ratio is smallest for small β, so Girolami's method chooses much larger π_i for the components which are "switched off", in that $\mathrm{E}_Q[a_i^2/\sigma^2] \approx 0$. It is thus more aggressively aiming for sparse solutions. In initial comparative experiments, Girolami's method outperformed SBL on the sparse linear model significantly in terms of the quality of inference approximation. Especially, the SBL marginal likelihood approximation turned out to be poor. A larger comparative study, from which conclusions can be drawn, is subject to future work.

A comparison between approximate inference techniques would be incomplete without including *variational mean field Bayes* (VMFB) (Attias, 2000; Ghahramani and Beal, 2001), maybe the most well known variational technique in the moment. It is also simply known as "variational Bayes" (see `www.variational-bayes.org`), although we understand this term as encompassing other variational methods for Bayesian inference as well, such as EP, SBL, direct site bounding, and others more. The distinctive feature of VMFB, previously known as "structured mean field", is the use of the generic mean field lower bound, as reviewed in Appendix D.2. VMFB for the sparse linear model is equivalent to direct site bounding, as has been shown in Palmer et al. (2006), and as is discussed in more detail in Appendix D.2. This equivalence holds as well for linear models with many other symmetric priors, for example Student's t.

8.3 Markov Chain Monte Carlo

While variational approximations are fairly established in machine learning, the dominant methods for approximating Bayesian inference in statistics are Markov chain Monte Carlo (MCMC) simulations (Neal, 1993; Gilks et al., 1996). In these techniques, a Markov chain over latent variables of interest (and possibly additional auxiliary ones) is simulated, whose stationary distribution is the desired posterior.

A simple MCMC method for the sparse linear model with Laplace prior has been proposed by Park and Casella (2005). They employ the scale mixture representation (9), introducing the scale parameters π as auxiliary variables alongside a. Their method is an instance of block Gibbs sampling, in that a is resampled given π, and vice versa. For simplicity, we denote the true posterior $P(\ldots|D)$ by Q in this section only. Now, the full conditional distribution $Q(a|\pi)$ is simply $N(a|h, \sigma^2\Sigma)$, with h, Σ defined as usual in terms of π (as in SBL above, $b = 0$ here), a Gaussian we can sample from easily (see Appendix B.2).

Next, the π_i are independent under $Q(\pi|a)$, with

$$Q(\pi_i|a) \propto \pi_i^{-3/2} \exp\left(\frac{-a_i^2\sigma^{-2}(\sqrt{2\lambda\sigma^2}/|a_i| - \pi_i)^2}{2\pi_i}\right) \propto \pi_i^{-3/2} \exp\left(\frac{-\tilde{\lambda}(\pi_i - \tilde{\mu})^2}{2\tilde{\mu}^2\pi_i}\right),$$

with $\tilde{\mu} = \sqrt{2\lambda\sigma^2}/|a_i|$, $\tilde{\lambda} = 2\lambda = \tau^2$. This density is the inverse Gaussian, which can be sampled from easily (see Chhikara and Folks, 1989, Section 4.5). The normalisation constant is $(\lambda/\pi)^{1/2}$.

Note that, just as with SBL and direct site bounding, we can use our existing EP code in order to implement this method as well. The π_i are resampled, instead of being updated deterministically. While they could be updated sequentially, Park and Casella (2005) consider joint updates which

tend to have better mixing properties. The representation, now maintaining the true $Q(a|\pi)$, has to be recomputed from scratch after each π update, so that each step costs $O(n \min\{m,n\}^2)$.

This sampler is certainly very simple to implement, especially with our representation code in place. Park and Casella (2005) give some arguments about the favourable role of log-concavity of $Q(a)$ for the sampler.[37] These are empirical, and even if good theoretical properties of MCMC samplers for log-concave posteriors have been established (Lovász and Vempala, 2003), these are different from the method considered here. Initial experiments with the sampler gave good results, although some erratic jumps in π components can be observed. The main cause of failure of block Gibbs samplers is the presence of strong dependencies between a and π. A more definite statement would require a comparison between this method and another sampler not based on scale variables.

The main advantage of MCMC over variational approximations is that it has no approximation bias in principle, if the chain is run for an unbounded amount of steps. In contrast, variational methods such as EP do have such a bias, which cannot be diminished by simply running them for longer.[38] It is also the case that simple variants of MCMC are typically fairly easy to implement, for example there are hardly ever problems with numerical stability. A main drawback of MCMC applied to problems of the sort considered here is that significantly more running time is required in order to obtain solutions of similar accuracy. Another major disadvantage is that a lot of expertise is required in order to run MCMC in a proper way. There are no convergence diagnostics which are easy to use or, in fact, are generally widely accepted. Most machine learning applications, such as the ones considered here, require methods which can be run robustly by users without extensive training in diagnosing Markov chain convergence. This problem becomes severe in the context of experiment design, where new decisions have to be done continuously, and even an expert would be hard pressed trying to diagnose proper convergence for all MCMC runs in between. Another drawback of MCMC is that while samples of the posterior are obtained, these cannot be used in a simple way in order to obtain a good estimate of the log marginal likelihood $\log P(u)$ (see Section 5). While the method of Chib (1995) proposes just that, it failed catastrophically in toy experiments of rather small scale with the sampler considered here, even if an excessive number of steps was used. This failure is interesting, given that the posterior is a log-concave (unimodal) distribution.

9. Discussion

We have shown how to perform accurate approximate Bayesian inference in the linear model with Laplace prior efficiently, by means of expectation propagation, and how this can be used to address tasks such as optimal design and hyperparameter estimation. The importance of numerical stability is raised for EP, and several means of improving robustness are proposed. Some implications of log-concavity for EP, and for approximate inference in general, have been shown.

The optimal design capability has been demonstrated for the application of gene regulatory network identification, where the sparsity prior was found to be essential in order to realise very significant gains. It is also motivated by preliminary experiments in the area of compressive sensing. Marginal likelihood optimisation has been used in order to optimise sparse codes for natural images,

37. They sample jointly over (a,σ^2), noting that $Q(a,\sigma^2)$ is log-concave in the transformation of (a,σ^2) described in Section 3.5.

38. Many variational methods allow for the choice of approximation families of varying complexity. For example, EP can be run with exponential families beyond the Gaussian, and even the case of Gaussian Q can potentially be improved by considering joint updates of blocks of sites. This requires the computation of multivariate non-Gaussian integrals, which is hard to do accurately, and is not done here.

in what constitutes an application of approximate inference on a large scale. Our experiments have been driven by a robust, efficient, and general implementation, which will be made available for scientific use.

9.1 Related Work

The sparse linear model (1) is of high practical relevance in statistics and machine learning, and has received a lot of attention. Some approximate inference techniques related to ours have been reviewed in Section 8. It is noted there that the computational representations and their robust update rules developed here, are required for these just as well.

The idea of L_1 regularisation of least squares has been used in very many contexts. The maximum a posteriori (MAP) treatment of the sparse linear model has been proposed as *Lasso* (Tibshirani, 1996) and as *basis pursuit* (Chen et al., 1999) (the latter for $\sigma^2 \to 0$). While the Lasso results in a quadratic program, basis pursuit is a linear programming problem. The prime advantage of an MAP treatment is that fitting to fixed data can be done very efficiently, in fact significantly faster than running EP until convergence. Very recently, several strong properties of the Lasso, basis pursuit, or other convex programming formulations of sparse estimation have been established, showing that in certain regimes they perfectly reconstruct very sparse signals in a minimax sense (Donoho and Elad, 2003; Candès et al., 2006; Wainwright, 2006). On the other hand, MAP as an approximation to Bayesian inference is fairly poor in this case. As noted in Section 3, a direct Laplace approximation is not well-defined for the sparse linear model. Even if this obvious problem was not present, the fact that there are many more variables than observations, renders the usual justification for Laplace's method obsolete. We have demonstrated a few advantages of going the full Bayesian way properly in this paper, such as optimal design based on uncertainty estimates, or marginal likelihood hyperparameter estimation. The MAP approximation for the sparse linear model has been applied to the gene network identification problem by Peeters and Westra (2004), but they did not address the problem of optimal design.

A general framework for EP on a class of hybrid models has been proposed by Zoeter and Heskes (2005). EP updates are done generically using Gaussian quadrature. Based on our findings here, EP for the sparse linear model with Laplace prior is very sensitive to the accuracy of EP updates, and the Gauss-Hermite rule would not lead to a working solution here. The generic proposal of converting between natural and moment parameterisation stated there is known to be unstable even in purely Gaussian models such as the Kalman filter, while our representation updates are essentially stable for log-concave sites. Also, the generality is quite restricted, in that they assume a fully factorised distribution family $\mathcal{F}$, which would not include joint Gaussians Q we consider here. Thus, while the prospect of a generic EP implementation is intriguing, important special cases such as Laplace or other sparsity prior sites, or joint Gaussian factors, would have to be treated as special cases. It remains to be seen whether the techniques to improve EP's numerical properties proposed here, are useful in this more general context as well.

Technically, our framework is quite related to the Independent Component Analysis method of Hojen-Sorensen et al. (2002), using Adaptive TAP (Opper and Winther, 2000) in order to estimate mean and covariance of the sources.[39] In fact, EP can be seen as particularly efficient way of searching for an ADATAP fixed point. They address the sparse image coding problem with the sparse linear model, but do not consider optimal design applications. Our approach is different to

39. What is meant is the *posterior* covariance of the sources, since in ICA, they are assumed to be independent *a priori*.

theirs in several important points. Their paper approaches a larger range of problems. On the other hand, they do not employ the natural EP marginal likelihood approximation we use here, but rather a variational bound. The study of Kuss and Rasmussen (2005) has indicated the superior quality of the EP approximation in a different, but related situation. Second, their image coding experiments are fairly small in scale, and they do not report any of the numerical problems we encountered, or in fact propose special measures to deal with such. Their paper treats numerically benign cases such as classification with logistic or probit likelihood alongside challenging (Laplace) or (in our opinion) highly problematic ones (Student's t; exponential power with exponent < 1), essentially recommending the same generic computations (which do not work, to the best of our knowledge and experience, in the situations we were interested in here).

9.2 Future Work

We have commented in Section 2.3 on the application of our method to the problem of learning and analysing image codes (Olshausen and Field, 1997; Lewicki and Olshausen, 1999), with the aim of understanding properties of visual neurons in the brain. In this context, the sparse linear model has been proposed as a useful setup, in which codes can be learned by maximising the marginal likelihood. The marginal likelihood approximation of Section 5 is more accurate than the one used by Lewicki and Olshausen (1999), and it will be interesting to test their hypothesis using our framework. A study with similar aims is given in Berkes et al. (2008), using variational mean field Bayes to approximate inference.

Other interesting applications lie in the area of compressive sensing. Some potential ones have been motivated in Section 2.4, and results will be reported in a later paper (Seeger and Nickisch, 2008). In this context, the large scale techniques motivated in Section 6 will be explored. Our preliminary findings in Section 7.3 indicate that approximate Bayesian inference and experimental design hold significant promises for compressive sensing, where so far approaches based on L_1-penalised estimation and random designs seem to predominate.

As detailed in Section 8, our EP framework is closely related to several other established methods of approximate inference. We plan to do a large, comparative study on several different tasks and data sets, where ground truth computations will be done via computationally intensive MCMC. We are not aware of existing comparative studies encompassing several approximate inference techniques for the sparse linear model.

Our experiences with the sparse linear model on the image coding problem (or with very underdetermined gene network identification settings) suggest that in some relevant cases, numerical stability issues seem to be inherently present in EP (i.e., are not just due to a bad implementation). These need to be understood much better, before we can seriously talk about generic EP solutions (Zoeter and Heskes, 2005), comparable to BUGS (Spiegelhalter et al., 1995) for Gibbs sampling or VIBES (Bishop and Winn, 2003) for variational mean field Bayes, both of which do not pose big problems of numerical stability. The sparse linear model seems a good test bed for such studies, different to the Gaussian process classification problem, which is numerically rather harmless. Since log-concavity helps in the important special case of fully Gaussian posterior approximations, its role needs to be understood better. Again, the Laplace prior of the sparse linear model will be important there, being "just about log-concave". Also, "cut-off" sites enforcing non-negativity, or more generally linear constraints (see Section 2.3), will play an important role there, not even being supported on all of $\mathbb{R}$.

In the MCMC approach of Park and Casella (2005), the noise variance σ^2 is integrated out along with the parameters a. This is possible (approximately) with EP as well, by choosing Q from an exponential family over (a, σ^2), which is not purely Gaussian. We have already done initial experiments with this extension, which will be reported in a later paper. In comparison to the method given here, the extension treats σ^2 as nuisance variable in a proper Bayesian fashion. It does not have to be chosen by other means, such as marginal likelihood maximisation. Much of the treatment of a, such as the representation of $Q(a|\sigma^2)$, or the analytical EP update w.r.t. a_i, is inherited from the framework given here. As an extension of EP beyond the case of fully Gaussian approximations, the extension is important as test bed for theoretical analyses. On the other hand, the extension is somewhat more complicated to implement, furthermore some of the numerical robustness of our method here is lost. For example, Theorem 1 does not hold for non-Gaussian Q. Moreover, the integration over σ^2 required by the EP updates cannot be done analytically, and approximate Gauss-Laguerre quadrature has to be used.

The Bayesian sparse linear model may have many other applications, given that its MAP variants (Lasso, basis pursuit) are very widely used. EP has also been applied to approximate inference in generalised linear models, where the likelihood is not Gaussian anymore, but comes from another exponential family. An application of this sparse generalised linear model to analysing neuronal spiking data is given in Seeger et al. (2007a) and Gerwinn et al. (2008), see also Qi et al. (2004). In this context, efficient online optimisation of experimental stimuli is an important task as well (Lewi et al., 2007).

The recent empirical success of EP in many different applications renders it important to gain a firm understanding of this technique. Some of the many relevant open questions are: For which models does the (single loop) EP algorithm provably converge? For which models is there no more than a single fixed point? How good is an EP approximation in terms of the marginals, and beyond that in terms of the covariance estimate? Numerical stability is an important issue for EP, which does not arise with most other approximate inference techniques. For which models can we expect numerical difficulties, and why? The step towards fractional EP may improve numerical properties of the method in general, but how do fractional EP approximations compare to the standard EP fixed points? Finally, how can EP fixed points be found for very large n, when the current practice of visiting each site in turn becomes unpractical?

Acknowledgments

The gene network identification application has been done in joint work with Florian Steinke and Koji Tsuda. The preliminary experiments with compressive sensing have been done in joint work with Hannes Nickisch. For the image coding application, we would like to acknowledge discussions with Matthias Bethge. We would like to thank Manfred Opper for interesting discussions about EP, and the anonymous referees for helpful comments. Supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

Appendix A. Details for the EP Update

In this section, we collect details concerning the EP update described in Section 3.3.

The Laplace site is $t_i(a) = \exp(-\tilde{\tau}|a|)$, $\tilde{\tau} = \tau/\sigma > 0$. Note that elsewhere, a prefactor of $\tilde{\tau}/2$ is used, which means that the value for Z_i to be computed here has to be multiplied post hoc.[40] We need to compute moments $I_k = \mathrm{E}_{N(h,\rho)}[a^k t_i(a)]$, $k = 0,1,2$, where we write $a = a_i$, $h = h_{\setminus i}$, $\rho = \rho_{\setminus i}$ for simplicity. W.l.o.g., we can assume that $\tilde{\tau} = 1$. Then, $I_0 = \tilde{I}_0(h) + \tilde{I}_0(-h)$, where some algebra gives

$$\tilde{I}_0(h) := \mathrm{E}\left[\mathrm{I}_{\{a\geq 0\}} e^{-a}\right] = \exp(\rho/2 - h)(1 - \Phi(\rho^{1/2} - h\rho^{-1/2})),$$

where Φ denotes the cumulative distribution function of $N(0,1)$. Now, from the definition of $\tilde{I}_0$, it is easy to see that $\tilde{I}_0(|h|) \geq \tilde{I}_0(-|h|)$, so that

$$\log I_0 = \log \tilde{I}_0(|h|) + \log\left(1 + \frac{\tilde{I}_0(-|h|)}{\tilde{I}_0(|h|)}\right)$$

can be computed in a stable manner. In the following, we make use of the well known asymptotic expansion

$$1 - \Phi(x) \sim N(x)x^{-1}\left(1 - 1x^{-2}\left(1 - 3x^{-2}\left(1 - 5x^{-2}\left(1 - 7x^{-2}(\ldots)\right)\right)\right)\right).$$

If $F(x) := \log(1 - \Phi(x))$, we use the asymptotic expansion up to $1 - 7x^{-2}$ for $x > 5$, while computing $F(x)$ exactly otherwise.[41] It is interesting to note that the simpler approximation $1 - \Phi(x) \approx N(x)/x$ is insufficient and leads to complete failure of EP on most tasks.

With this in mind, we have that $\log \tilde{I}_0(|h|) = \rho/2 - |h| + F(\rho^{1/2} - |h|\rho^{-1/2})$, and

$$R := \frac{\tilde{I}_0(-|h|)}{\tilde{I}_0(|h|)} = \exp\left(2|h| + F(\rho^{1/2} + |h|\rho^{-1/2}) - F(\rho^{1/2} - |h|\rho^{-1/2})\right).$$

For example, if $\rho^{1/2} - |h|\rho^{-1/2} > 5$, we use the tail approximation for both F terms. Namely, if the approximation is $1 - \Phi(x) \approx N(x)x^{-1}g(x)$, we end up with

$$R = \frac{(\rho - |h|)g(\rho^{1/2} + |h|\rho^{-1/2})}{(\rho + |h|)g(\rho^{1/2} - |h|\rho^{-1/2})}.$$

Note that in general, $R \in (0,1]$.

Next, $I_1 = \tilde{I}_1(h) - \tilde{I}_1(-h)$, with

$$\tilde{I}_1(h) := \mathrm{E}\left[\mathrm{I}_{\{a\geq 0\}} a e^{-a}\right] = (h - \rho)\tilde{I}_0(h) + \rho^{1/2}\exp(\rho/2 - h)\mathrm{E}\left[\mathrm{I}_{\{s\geq \rho^{1/2} - h\rho^{-1/2}\}} s\right],$$

where $s \sim N(0,1)$. Using $(dN(s))/ds = -sN(s)$, we have that $\mathrm{E}[\mathrm{I}_{\{s\geq s_0\}} s] = N(s_0)$. Furthermore,

$$\exp(\rho/2 \pm h)N(\rho^{1/2} \pm h\rho^{-1/2}) = N(h\rho^{-1/2}),$$

which does not depend on $\operatorname{sgn} h$. Therefore, the mean of $\hat{P}_i(a)$ is

$$\begin{aligned}\hat{h} = \frac{I_1}{I_0} &= \frac{(h - \rho)\tilde{I}_0(h) - (-h - \rho)\tilde{I}_0(-h)}{I_0} = h + \rho\frac{\tilde{I}_0(-h) - \tilde{I}_0(h)}{I_0} \\ &= h + \rho(\operatorname{sgn} h)\left(1 - 2(1 + R)^{-1}\right).\end{aligned}$$

40. The reason for dropping this prefactor is that we want to deal with the case of fractional sites (see Section 3.3.1) t_i^η by simply replacing τ by $\eta\tau$.

41. The C math library provides `log1p`$(x) = \log(1 + x)$, which is accurate for small $|x|$.

Since $I_1/I_0 = h + \rho\beta_i$, we have that

$$\beta_i = (\mathrm{sgn}\, h)\left(1 - 2(1+R)^{-1}\right). \tag{11}$$

Next, $I_2 = \tilde{I}_2(h) + \tilde{I}_2(-h)$, where some algebra gives

$$\begin{aligned}
\tilde{I}_2(h) &:= \mathrm{E}\left[\mathrm{I}_{\{a\geq 0\}} a^2 e^{-a}\right] \\
&= (\rho^2 - h^2)\tilde{I}_0(h) + 2h\tilde{I}_1(h) - 2\rho^{3/2} N(h\rho^{-1/2}) + \rho\exp(\rho/2 - h)\mathrm{E}\left[\mathrm{I}_{\{s\geq \rho^{1/2} - h\rho^{-1/2}\}} s^2\right] \\
&= (h-\rho)^2\tilde{I}_0(h) + 2\rho^{1/2} N(h\rho^{-1/2})(h-\rho) + \rho\exp(\rho/2 - h)\mathrm{E}\left[\mathrm{I}_{\{s\geq \rho^{1/2} - h\rho^{-1/2}\}} s^2\right].
\end{aligned}$$

Using $s^2 N(s) = N(s) - (dsN(s))/ds$, we see that

$$\rho\exp(\rho/2 - h)\mathrm{E}\left[\mathrm{I}_{\{s\geq \rho^{1/2} - h\rho^{-1/2}\}} s^2\right] = \rho\tilde{I}_0(h) + \rho(\rho^{1/2} - h\rho^{-1/2}) N(h\rho^{-1/2}).$$

Together, we have

$$\tilde{I}_2(h) = (h^2 + \rho^2 + \rho - 2h\rho)\tilde{I}_0(h) + \rho^{1/2} N(h\rho^{-1/2})(h-\rho),$$

thus

$$I_2 = (h^2 + \rho^2 + \rho) I_0 - 2\rho h(\tilde{I}_0(h) - \tilde{I}_0(-h)) - 2\rho^{3/2} N(h\rho^{-1/2}).$$

Using that $\beta_i = (\tilde{I}_0(-h) - \tilde{I}_0(h))/I_0$ and $\hat{h} = h + \rho\beta_i$, some algebra gives that

$$\frac{I_2}{I_0} = \rho + \rho^2 + h^2 + 2h(\hat{h} - h) - 2\rho^{3/2} N(h\rho^{-1/2}) I_0^{-1}.$$

Therefore, the variance of $\hat{P}_i(a)$ is

$$\begin{aligned}
\hat{\rho} &= \frac{I_2}{I_0} - \hat{h}^2 = -h^2 + (2h - \hat{h})\hat{h} + \rho + \rho^2 - 2\rho^{3/2} N(h\rho^{-1/2}) I_0^{-1} \\
&= \rho + \rho^2(1 - \beta_i^2) - 2\rho^{3/2} N(h\rho^{-1/2}) I_0^{-1},
\end{aligned}$$

since $(2h - \hat{h})\hat{h} = h^2 - (\rho\beta_i)^2$. Since $\hat{\rho} = \rho(1 - \rho\nu_i)$, we have that

$$\nu_i = \beta_i^2 - 1 + (\pi\rho/2)^{-1/2} \exp\left(-\frac{h^2}{2\rho} - \log I_0\right).$$

Finally, in order to incorporate $\tilde{\tau} \neq 1$, we note that this simply means plugging in $h = \tilde{\tau} h_{\setminus i}$, $\rho = \tilde{\tau}^2 \rho_{\setminus i}$ above, and multiplying β_i by $\tilde{\tau}$, ν_i by $\tilde{\tau}^2$. Note that $Z_i = I_0 = \mathrm{E}_{Q^{\setminus i}}[t_i(a_i)]$ is not required for the EP update itself, but has to be evaluated if an approximation to the marginal likelihood $P(D)$ is sought (see Section 5; recall that Z_i as computed here has to be multiplied with the prefactor $\tilde{\tau}/2$ of t_i which we omitted).

A.1 The Role of Log-concavity

In this section, we give the proof of Theorem 1. Recall the definition of log-concavity and the marginalisation theorem of Prékopa from Section 3.5. For an update at site i, we can assume that $Q^{\setminus i}(a_i)$ is a proper Gaussian. We begin by showing that $Z_i = \mathrm{E}_{Q^{\setminus i}}[t_i(a_i)]$ is log-concave in $h_{\setminus i}$. Namely, $\log Q^{\setminus i}$ is jointly concave in $(a_i, h_{\setminus i})$ (being a negative quadratic in $a_i - h_{\setminus i}$), so that $t_i(a_i)Q^{\setminus i}(a_i|h_{\setminus i})$ is log-concave in $(a_i, h_{\setminus i})$. Then, $Z_i(h_{\setminus i})$ is log-concave by the marginalisation theorem. Therefore, $\nu_i = -(\partial^2 \log Z_i)/(\partial h_{\setminus i}^2) \geq 0$ (see Section 3.3). The variance of $\hat{P}_i$ is $\sigma^2\rho_i'$, where $\rho_i' = \rho_{\setminus i}(1 - \sigma^2\nu_i\rho_{\setminus i})$. Since t_i is bounded with support of positive measure, this variance exists and is positive, implying that $1 - \sigma^2\nu_i\rho_{\setminus i} \in (0, 1]$. But $\pi_i' = \sigma^2\nu_i/(1 - \sigma^2\nu_i\rho_{\setminus i}) \geq \sigma^2\nu_i \geq 0$, so π_i' remains nonnegative throughout.

Appendix B. Details for Sequential Design

In this section, we collect details for the sequential design application of the sparse linear model.

B.1 The Simple Information Gain Score

The simple information gain is introduced in Section 4.1. Recall the Gaussian relative entropy from (4), and the fact that $M = I + x_* x_*^T \Sigma$. Thus, if $\alpha := 1 + x_*^T \Sigma x_*$, then $\log|M| = \log\alpha$, using the relation $|I + VW^T| = |I + W^T V|$. Furthermore, the Woodbury formula (Henderson and Searle, 1981) gives $M^{-1} = I - \alpha^{-1} x_* x_*^T \Sigma$, so that $\mathrm{tr}(M^{-1} - I) = \alpha^{-1} - 1$.

Finally, $\tilde{b}' = \tilde{b} + u_* x_*$, where $\tilde{b} = b^{(0)} + b$ (see Section 3.4), so that

$$h' = \left(\Sigma - \alpha^{-1}\Sigma x_* x_*^T \Sigma\right)\left(\tilde{b} + u_* x_*\right) = h + \alpha^{-1}(u_* - x_*^T h)\Sigma x_*,$$

and

$$(h' - h)^T \Sigma^{-1}(h' - h) = (\alpha - 1)\alpha^{-2}(u_* - x_*^T h)^2.$$

Altogether, the simple information gain score is

$$S(x_*, u_*) = \frac{1}{2}\left(\log\alpha + \frac{\alpha - 1}{\alpha}\left(-1 + \alpha^{-1}\left(\frac{u_* - x_*^T h}{\sigma}\right)^2\right)\right).$$

We need to compute α and $x_*^T h$. In the degenerate case, let $v = L^{-1} X \Pi^{-1} x_*$, then $\alpha = 1 + x_*^T \Pi^{-1} x_* - \|v\|^2$, and $x_*^T h = x_*^T \Pi^{-1}(b^{(0)} + b) - v^T\gamma$. In the non-degenerate case, let $v = L^{-1} x_*$, then $\alpha = 1 + \|v\|^2$, and $x_*^T h = v^T\gamma$.

The marginal criteria of Section 4.2 require the computation of $z_* = \Sigma x_*$. In the non-degenerate case, $z_* = L^{-T} v$. In the degenerate case, $z_* = \Pi^{-1}(x_* - X^T L^{-T} v)$.

B.2 Sampling A

We need to sample from $Q(A|D)$ in order to approximate the expected information gain, as noted in Section 4. Let $Q(a)$ be the posterior over a row of A, based on the representation given in Section 3.2, and let $n \sim N(0, I)$. In the non-degenerate case, $a = L^{-T}(\sigma n + \gamma)$ is distributed according to $N(h, \sigma^2\Sigma)$.

Sampling is more difficult in the degenerate case. Let

$$I + X\Pi^{-1}X^T = UDU^T$$

be the spectral decomposition, where D is diagonal and nonnegative, and $U \in \mathbb{R}^{m,m}$ is orthonormal. We make the ansatz

$$c = \left(I - \Pi^{-1} X^T U R U^T X\right) \Pi^{-1/2} n$$

with diagonal R. $\mathrm{E}[cc^T] = \Sigma$ gives $(D-I)R^2 - 2R + D^{-1} = 0$, which is solved by $R = \mathrm{diag}(1/(\sqrt{d_i}(\sqrt{d_i}+1)))_i$. Finally, $a = \sigma c + h$.

Appendix C. The Marginal Likelihood

In this section, we derive the EP marginal likelihood approximation and its gradient w.r.t. model parameters. Recall the discussion of Section 5, the definition of L is given in (6). First, $\log C_i = \eta^{-1}(\log Z_i - \log \tilde{Z}_i)$. The computation of $\log Z_i$ is discussed in Appendix A. Some algebra (Seeger, 2005) gives

$$\log \tilde{Z}_i = \frac{1}{2}\left(\log(1-\eta\pi_i\rho_i) - \frac{\eta\pi_i h_i^2 - 2h_i\eta b_i + \rho_i(\eta b_i)^2}{\sigma^2(1-\eta\pi_i\rho_i)}\right),$$

where $Q(a_i) = N(a_i|h_i, \sigma^2\rho_i)$.

We begin with $\nabla_X L$ and $\partial L/\partial\sigma^{-2}$ (both parameters of $P^{(0)}$), using (7). Since σ^2 also features explicitly in the sites t_i, the derivative is the sum of two parts, and we deal with the second part below.

$$d\log P^{(0)}(a) = \mathrm{tr}\left(\sigma^{-2} e a^T\right)^T (dX) + \frac{1}{2}\left(m\sigma^2 - \|e\|^2\right)(d\sigma^{-2}), \quad e := u - Xa.$$

If $Q(a) = N(h, \sigma^2\Sigma)$, then

$$\mathrm{E}_Q\left[d\log P^{(0)}(a)\right] = \mathrm{tr}\left(\sigma^{-2} f h^T - X\Sigma\right)^T (dX) - \frac{1}{2}\left(\|f\|^2 + \sigma^2\,\mathrm{tr}\, X\Sigma X^T - m\sigma^2\right)(d\sigma^{-2}),$$
$$f := \mathrm{E}_Q[e] = u - Xh.$$

Now, $\mathrm{tr}\, X\Sigma X^T = \mathrm{tr}(I - \Sigma\Pi) = n - (\mathrm{diag}\,\Sigma)^T\pi$, so that

$$dL = \mathrm{tr}\left(\sigma^{-2} f h^T - X\Sigma\right)^T (dX) - \frac{1}{2}\left(\|f\|^2 - \sigma^2(\mathrm{diag}\,\Sigma)^T\pi + (n-m)\sigma^2\right)(d\sigma^{-2}).$$

The derivative w.r.t. $\tilde{\tau} = \tau/\sigma$ is computed using (8). We have that $(d/d\tilde{\tau})\log t_i(a_i) = -|a_i| + 1/\tilde{\tau}$, so we need to compute

$$\mathrm{E}_{\hat{P}_i}[-|a_i|] = -Z_i^{-1}\mathrm{E}_{Q^{\setminus i}}[|a_i| t_i(a_i)],$$

which is of similar from to I_1 in Appendix A. In the notation used there, if $\hat{I}_1 = \tilde{I}_1(h) + \tilde{I}_1(-h)$, then $\hat{I}_1/I_0 = -\rho - \beta_i h + 2\rho^{1/2} N(h\rho^{-1/2}) I_0^{-1}$. Plugging in $h = \tilde{\tau} h_{\setminus i}$, $\rho = \tilde{\tau}^2\rho_{\setminus i}$, and dividing by $\tilde{\tau}$, we have that

$$-Z_i^{-1}\mathrm{E}_{Q^{\setminus i}}[|a_i| t_i(a_i)] = \tilde{\tau}\rho_{\setminus i} + \beta_i h_{\setminus i} - 2\rho_{\setminus i}^{1/2} N(h_{\setminus i}\rho_{\setminus i}^{-1/2}) I_0^{-1},$$

where β_i is given by (11) (it is not multiplied by $\tilde{\tau}$). Finally, $d\tilde{\tau} = \sigma^{-1}(d\tau) + \frac{1}{2}\tau\sigma(d\sigma^{-2})$, whereby we can complete the derivative w.r.t. σ^{-2} as well.

As an aside, there is a subtle issue concerning the derivative w.r.t. σ^2. Seeger (2005) shows that indirect dependencies on hyperparameters through the site parameters do not have to be taken into account when computing the gradient. But if the derivative of (6) w.r.t. σ^2 is computed, keeping

b_i, π_i constant, the result is different from ours here. This is explained by our non-standard parameterisation of site parameters in the present paper. Namely, what is referred to as site parameters in Seeger (2005), are in fact the $\sigma^{-2}b_i, \sigma^{-2}\pi_i$ here, *not* b_i, π_i. If the former are kept constant, a direct differentiation of (6) renders our result here.

Appendix D. Related Approximate Inference Techniques

In this section, we give details on the approximate inference techniques discussed in Section 8.

D.1 Sparse Bayesian Learning

Recall Section 8.1. In order to compute the marginal likelihood $P(u,\pi)$, we note that

$$P^{(0)}(a)N(a|0,\sigma^2\Pi^{-1}) = (2\pi\sigma^2)^{-m/2}e^{-(\sigma^{-2}/2)\|u\|^2}(2\pi\sigma^2)^{-n/2}|\Pi|^{1/2} N^U(a|\sigma^{-2}b^{(0)},\sigma^{-2}(X^TX+\Pi)),$$

so that with $h = \Sigma b^{(0)}$, $\Sigma^{-1} = X^TX + \Pi$, some algebra gives

$$2\log P(u,\pi) = \sigma^{-2}h^Tb^{(0)} + \log|\Sigma| - \tau^2 1^T(\pi^{-1}) + C,$$

where $C = -3\log|\Pi| - m\log(2\pi\sigma^2) - \sigma^{-2}\|u\|^2 + 2n\log(\tau^2/2)$.

We need to maximise $\log P(u,\pi)$ w.r.t. π_i, keeping all other π_j fixed. Then, $d\Pi = (d\pi_i)\delta_i\delta_i^T$, and let $\log P(u,\pi) = (1/2)\psi + C$. Furthermore, $Q(a_i) = N(a_i|h_i,\sigma^2\rho_i)$, that is, $\rho_i = \Sigma_{i,i}$. Now, $d\log|\Pi + X^TX| = \rho_i(d\pi_i)$, and $d\sigma^{-2}h^Tb^{(0)} = -\sigma^{-2}h_i^2(d\pi_i)$, so that

$$d\psi = \left(-\sigma^{-2}h_i^2 - 3\pi_i^{-1} - \rho_i + \frac{\tau^2}{\pi_i^2}\right)d\pi_i.$$

Equating this to zero results in a quadratic equation for π_i, whose nonnegative solution is given by (10).

D.2 Variational Mean Field Bayes

The *variational mean field Bayesian* (VMFB) framework is a fairly generic approach to variational inference. It starts from the classical variational characterisation of inference (Wainwright and Jordan, 2003):

$$\log P(u) = \sup_Q \mathrm{E}_Q\left[\log P(u,a,\sigma^2,\pi) - \log Q(a,\sigma^2,\pi)\right], \tag{12}$$

then relaxes the problem by imposing factorisation constraints on allowable Q (the optimal unconstrained choice for Q is the true posterior).[42] The variational characterisation is also known as mean field lower bound, because it is the defining feature of (structured) mean field approximations (Jordan et al., 1997).

Once appropriate factorisation assumptions are placed on Q, the feasible set can be written analytically in terms of factors from these families, and the right hand side of (12) and its gradient

42. Here, we introduce the scale parameters π_i by employing the scale mixture representation (9). VMFB works for models which can be represented exclusively in terms of exponential family distributions, which is often possible by introducing latent variables. One could possibly choose another representation of the Laplace sites, whence the equivalence of VMFB and direct site bounding would not hold, but this is not done here.

can be computed easily. Furthermore, this expression now lower bounds $\log P(u)$, because the maximisation is over the subset of factorising Q. On the other hand, the optimisation over factorising Q is not convex in general, and usually only a local optimum is found. Moreover, even the global maximum is the minimiser of $\mathrm{D}[Q \,\|\, P(\cdot|D)]$ over factorising Q (this is also the slack in the lower bound), so that Q does not in general have the same marginal moments as $P(\cdot|D)$. The latter would be obtained by minimising $\mathrm{D}[P(\cdot|D) \,\|\, Q]$ over factorising Q, but not even local minima of the latter can be found by any tractable method currently known.

For fixed σ^2, it has been shown in Palmer et al. (2006) that VMFB is strongly equivalent to Girolami's method of Section 8.2, in that the variational parameters, their updates, and the $\log P(u)$ lower bound are the same. We make the factorisation assumption $Q(a,\pi) = Q(a)Q(\pi)$. The resulting lower bound on $\log P(u)$ is optimised by updating the factors in turn, fixing the corresponding other one. If we fix $Q(\pi)$, the maximiser is $Q(a) = N(h, \sigma^2\Sigma)$, where h, Σ are defined as usual, but plugging in $\mathrm{E}_Q[\pi]$ for π. If $Q(a)$ is kept fixed, then the maximiser is

$$Q(\pi) \propto e^{\mathrm{E}_{Q(a,\sigma^2)}[\log P(\pi|a,\sigma^2,u)]} \propto P(\pi) e^{\mathrm{E}_{Q(a,\sigma^2)}[\log P(a|\sigma^2,\pi)]},$$

which decomposes w.r.t. the π_i. The form is given in Section 8.3, namely $\log Q(\pi_i) = C + \log \pi_i^{-3/2} - (\pi_i^2 \mathrm{E}[a_i^2\sigma^{-2}] + 2\lambda)/(2\pi_i)$, which is inverse Gaussian with mean $\tilde{\mu} = \tau/\sqrt{\mathrm{E}[a_i^2\sigma^{-2}]}$ and $\tilde{\lambda} = \tau^2$. A sequential VMFB variant iterates over the sites, updating $\pi_i' = \tau/\sqrt{\mathrm{E}[a_i^2\sigma^{-2}]}$. This is algorithmically equivalent the direct site bounding method of Section 8.2.

Appendix E. Modifications of Olshausen/Field Code

We compare our method against the one proposed by Olshausen and Field (1997), using their code which can be obtained at `http://redwood.berkeley.edu/bruno/sparsenet/`. Since the code is written for interactive use, we had to modify it in order to work for our study, which compares fully automatic methods.

First, our modification accepts a fixed training set of $r = 50000$ image patches, while the original code extracts patches on the fly.[43] A sweep over the whole set consists of 500 batch updates, where batches are drawn at random without replacement. 20 sweeps are done in total.

The code comes with several parameters. A study of the code reveals that `noise_var` is our σ^2, `beta` is our $\tilde{\tau} = \tau/\sigma$, and `sigma` is set to one. There is a learning rate parameter `eta`, which the documentation recommends to set by hand, starting with $\eta = 5$, reducing it towards $\eta = 1$ (the default value in the code). We chose the following schedule: $\eta = 5, 4, 3$ for 100, $\eta = 2$ for 500, then $\eta = 1$ for the remaining 9200 updates.

References

H. Attias. A variational Bayesian framework for graphical models. In S. Solla, T. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000.

43. We used their extraction code in order to create the data set in the first place.

P. Berkes, R. Turner, and M. Sahani. On sparsity and overcompleteness in image models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

C. Bishop and J. Winn. Structured variational distributions in VIBES. In C. Bishop and B. Frey, editors, *Workshop on Artificial Intelligence and Statistics 9*, pages 244–251, 2003. Electronic Proceedings (ISBN 0-9727358-0-1).

V. Bogachev. *Gaussian Measures*. Mathematical Surveys and Monographs. American Mathematical Society, 1998.

L. Bottou. Online learning and stochastic approximations. In D. Saad, editor, *On-Line Learning in Neural Networks*. Cambridge University Press, 1998.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2002.

E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52 (2):489–509, 2006.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3): 273–304, 1995.

S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

R. Chhikara and L. Folks. *The Inverse Gaussian Distribution: Theory, Methodology, and Applications*. Marcel Dekker Inc., 1989.

S. Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90(432):1313–1321, 1995.

J. DeRisi, V. Iyer, and P. Brown. Exploring the matebolic and genetic control of gene expression on a genomic scale. *Science*, 282:699–705, 1997.

J. Dongarra, C. Moler, J. Bunch, and G. Stewart. *LINPACK User's Guide*. Society for Industrial and Applied Mathematics, 1979.

D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

D. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proc. Natl. Acad. Sci. USA*, 100:2197–2202, 2003.

A. Faul and M. Tipping. Analysis of sparse Bayesian learning. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 383–389. MIT Press, 2002.

V. Fedorov. *Theory of Optimal Experiments*. Academic Press, 1972.

M. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1050–1059, 2003.

T. Gardner, C. Cantor, and J. Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767):339–342, 2000.

S. Gerwinn, J. Macke, M. Seeger, and M. Bethge. Bayesian inference for spiking neuron models with a sparsity prior. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 507–513. MIT Press, 2001.

W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1st edition, 1996.

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41(2): 337–348, 1992.

M. Girolami. A variational method for learning sparse and overcomplete representations. *Neural Computation*, 13:2517–2532, 2001.

T. Gneiting. Normal scale mixtures and dual probability densities. *J. Statist. Comput. Simul.*, 59: 375–384, 1997.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

H. Henderson and S. Searle. On deriving the inverse of a sum of matrices. *SIAM Review*, 23:53–60, 1981.

P. Hojen-Sorensen, O. Winther, and L. Hansen. Mean field approaches to independent component analysis. *Neural Computation*, 14:889–918, 2002.

R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1st edition, 1985.

H. Ishwaran and J. Rao. Spike and slab gene selection for multigroup microarray data. *Journal of the American Statistical Association*, 100(471):764–780, 2005.

T. Jaakkola. *Variational Methods for Inference and Estimation in Graphical Models*. PhD thesis, Massachusetts Institute of Technology, 1997.

S. Ji and L. Carin. Bayesian compressive sensing and projection optimization. In Z. Ghahramani, editor, *International Conference on Machine Learning 24*. Omni Press, 2007.

M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods in graphical models. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1997.

B. Kholodenko, A. Kiyatkin, F. Bruggeman, E. Sontag, H. Westerhoff, and J. Hoek. Untangling the wires: A strategy to trace functional interactions in signaling and gene networks. *PNAS*, 99(20): 12841–12846, 2002.

H. Kushner and A. Budhiraja. A nonlinear filtering algorithm based on an approximation of the conditional distribution. *IEEE Transactions on Automatic Control*, 45:580–585, 2000.

M. Kuss and C. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 609–616. MIT Press, 2003.

J. Lewi, R. Butera, and L. Paninski. Real-time adaptive information-theoretic optimization of neurophysiological experiments. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

M. Lewicki and B. Olshausen. Probabilistic framework for the adaption and comparison of image codes. *J. Opt. Soc. Amer. A*, 16(7):1587–1601, 1999.

L. Lovász and S. Vempala. Hit and run is fast and fun. Technical Report MSR-TR-2003-05, Microsoft Research, Redmond, January 2003.

D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):589–603, 1991.

D. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, January 2001a.

T. Minka. Expectation propagation for approximate Bayesian inference. In J. Breese and D. Koller, editors, *Uncertainty in Artificial Intelligence 17*. Morgan Kaufmann, 2001b.

T. Minka. Power EP. Technical report, Microsoft Research, Cambridge, 2004.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993. See `www.cs.toronto.edu/~radford`.

R. M. Neal. *Bayesian Learning for Neural Networks*. Number 118 in Lecture Notes in Statistics. Springer, 1996.

A. O'Hagan. *Bayesian Inference*, volume 2B of *Kendall's Advanced Theory of Statistics*. Arnold, London, 1994.

B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

M. Opper and O. Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6:2177–2204, 2005.

M. Opper and O. Winther. Gaussian processes for classification: Mean field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

A. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao. Variational EM algorithms for non-Gaussian latent variable models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, 2006.

L. Paninski. Log-concavity results on Gaussian process methods for supervised and unsupervised learning. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

T. Park and G. Casella. The Bayesian Lasso. Technical report, University of Florida, 2005.

R. Peeters and R. Westra. On the identification of sparse gene regulatory networks. In *Proc. 16th Int. Symp. on Math. Theory of Networks*, 2004.

J. Pratt. Concavity of the log likelihood. *Journal of the American Statistical Association*, 76(373): 103–106, 1981.

Y. Qi, T. Minka, R. Picard, and Z. Ghahramani. Predictive automatic relevance determination by expectation propagation. In C. Brodley, editor, *International Conference on Machine Learning 21*. Morgan Kaufmann, 2004.

L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1st edition, 2003.

S. Rogers and M. Girolami. A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, 21(14):3131–3137, 2005.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. International Thomson Publishing, 1st edition, 1996.

M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, July 2003. See `www.kyb.tuebingen.mpg.de/bs/people/seeger`.

M. Seeger. Low rank updates for the Cholesky decomposition. Technical report, University of California at Berkeley, 2004. See `www.kyb.tuebingen.mpg.de/bs/people/seeger`.

M. Seeger. Expectation propagation for exponential families. Technical report, University of California at Berkeley, 2005. See `www.kyb.tuebingen.mpg.de/bs/people/seeger`.

M. Seeger and H. Nickisch. Compressed sensing and Bayesian experimental design. To appear at ICML, 2008.

M. Seeger, S. Gerwinn, and M. Bethge. Bayesian inference for sparse generalized linear models. In J. Kok, J. Koronacki, R. Lopez, S. Matwin, D. Mladenic, and A. Skowron, editors, *European Conference on Machine Learning 18*. Springer, 2007a.

M. Seeger, F. Steinke, and K. Tsuda. Bayesian inference and optimal design in the sparse linear model. In M. Meila and X. Shen, editors, *Workshop on Artificial Intelligence and Statistics 11*, 2007b.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Conference on Computational Learning Theory 5*, pages 287–294. Morgan Kaufmann, 1992.

D. Spiegelhalter, A. Thomas, N. Best, and W. Gilks. BUGS: Bayesian inference using Gibbs sampling. Technical report, MRC Biostatistics Unit, Cambridge University, 1995.

F. Steinke, M. Seeger, and K. Tsuda. Experimental design for efficient identification of gene regulatory networks using sparse Bayesian models. *BMC Systems Biology*, 1(51), 2007.

J. Tegnér, M. Yeung, J. Hasty, and J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *PNAS*, 100(10):5944–5949, 2003.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of Roy. Stat. Soc. B*, 58: 267–288, 1996.

M. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

M. J. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using l_1-constrained quadratic programming. Technical Report 709, UC Berkeley, Dept. of Statistics, 2006.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, UC Berkeley, Dept. of Statistics, 2003.

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

D. Wipf, J. Palmer, B. Rao, and K. Kreutz-Delgado. Performance analysis of latent variable models with sparse priors. In *Proceedings of ICASSP 2007*, 2007.

O. Zoeter and T. Heskes. Gaussian quadrature based expectation propagation. In Z. Ghahramani and R. Cowell, editors, *Workshop on Artificial Intelligence and Statistics 10*, 2005.