Bayesian Inference and Optimal Design in the Sparse Linear Model

Matthias Seeger
Max Planck Institute for
Biological Cybernetics
Tübingen, Germany
seeger@tuebingen.mpg.de

Florian Steinke
Max Planck Institute for
Biological Cybernetics
Tübingen, Germany
steinke@tuebingen.mpg.de

Koji Tsuda
Max Planck Institute for
Biological Cybernetics
Tübingen, Germany
koji.tsuda@tuebingen.mpg.de

Abstract

The sparse linear model has seen many successful applications in Statistics, Machine Learning, and Computational Biology, such as identification of gene regulatory networks from micro-array expression data. Prior work has either approximated Bayesian inference by expensive Markov chain Monte Carlo, or replaced it by point estimation. We show how to obtain a good approximation to Bayesian analysis efficiently, using the Expectation Propagation method. We also address the problems of optimal design and hyperparameter estimation. We demonstrate our framework on a gene network identification task.

1 Introduction

We consider the *linear model*

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

where $\boldsymbol{X} \in \mathbb{R}^{m,n}$ is the design matrix, $\boldsymbol{u} \in \mathbb{R}^m$ the reponses, and $\boldsymbol{a} \in \mathbb{R}^n$ is the vector of parameters (or weights). $\boldsymbol{X}$, $\boldsymbol{u}$ are being observed. In the applications we consider, sparsity of $\boldsymbol{a}$ is a key prior assumption: elements of $\boldsymbol{a}$ should be set to zero, whenever they are not required to describe the data well. On the other hand, elements which are required, should be allowed to be big if necessary. Many sparsity-favouring priors have been suggested in Statistics. In this paper, we concentrate on *Laplace distribution* priors of the form

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

A key advantage of this choice over others is log-concavity, which implies important computational advantages (more details about this point are given in [16]). We refer to the linear model with a Laplace prior as *sparse linear model*. It is important to note that the linear model is typically employed with a *Gaussian* prior $P(\boldsymbol{a})$, which due to conjugacy allows for simple analytic treatment (see [9], Chap. 9). However, such a prior fails to encode sparsity[1] as a property of $\boldsymbol{a}$, justifying the complications of using a Laplace prior (see also Section 6).

In this paper, we show how to perform accurate approximate Bayesian inference for $\boldsymbol{a}$ in the sparse linear model, using the Expectation Propagation (EP) algorithm. One of our main interests here is in the area of optimal design (see Section 2), where decisions have to be taken based on very few observations. To this end, the ability of encoding prior knowledge and valid estimation of uncertainty are vital, and these come natural in a Bayesian framework. While Bayesian inference can be performed using Markov chain Monte Carlo [12], our approach is much more efficient and can be applied to large problems of interest in Machine Learning. The application of EP to the sparse linear model proves challenging, due to the underdetermined nature of the likelihood, and some novel techniques are introduced here in order to obtain a numerically stable algorithm.

The structure of this paper is as follows. In Section 2, a key application of the sparse linear model is 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 present experimental results in Section 6, Section 7 refers to related work and concludes the paper.

Efficient and extendable code for the sparse linear model will be put in the public domain. Some details had to be omitted from this short paper, they can be found in the longer report [16]. For example, there are two regimes which require different treatments: the de-

[1] See [19] for a good discussion of this point.

generate where $n > m$, and the non-degenerate with $n \leq m$. The former is harder to deal with and of prime interest in the context here, so details for the latter are omitted here and are given in [16].

2 Gene Network Identification

Measuring m-RNA expression levels for many genes in parallel is affordable and widely done today using DNA micro-arrays [3]. One goal of such efforts is to recover regulatory networks. For example, some genes may code for transcription factor proteins, which up-/downregulate 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 linearized at its steady state:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{u}(t) + \boldsymbol{\varepsilon}(t),$$

where $\boldsymbol{x}(t)$ is the deviation in expression from steady state, and $\boldsymbol{\varepsilon}(t)$ is white noise. $\boldsymbol{A} \in \mathbb{R}^{n,n}$ is the system matrix, whose non-zero entries represent the edges of the network. Note that $\boldsymbol{A}$ is square, and for a stable steady state, it is nonsingular. $\boldsymbol{u}(t)$ is an external control, allowing the active user to probe the unknown $\boldsymbol{A}$. It is generally assumed that $\boldsymbol{u}(t)$ is small enough not to drive the system out of its linearity region. Due to the very noisy environment, it is typical to restrict controls to be constant, $\boldsymbol{u}(t) \equiv \boldsymbol{u}$, and to measure the new steady state $\lim_{t\to\infty} \boldsymbol{x}(t)$ [17]. Such disturbances $\boldsymbol{u}$ may be implemented biologically using gene switches [4], which puts further restrictions on allowable controls.

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

$$P(\boldsymbol{a}^{(j)}|D) \propto N(\boldsymbol{U}_{\cdot,j}|\boldsymbol{X}\boldsymbol{a}^{(j)}, \sigma^2\boldsymbol{I}) \prod_i P(a_{j,i}).$$

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

Since biological experiments involving gene switches are very expensive, a key requirement is to perform with as few data as possible, which is possible if biological prior knowledge is encoded in $P(\boldsymbol{A})$. Importantly, regulatory networks are observed to be sparsely connected, *i.e.* plausible $\boldsymbol{A}$ 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 $\boldsymbol{u}_*$, which of these experiments renders most new information about $\boldsymbol{A}$? Thus, a "value of information" is sought which can be computed for each candidate $\boldsymbol{u}_*$ *without* doing the corresponding experiment. Optimal design is well developed within Bayesian analysis [6], and access to this methodology is a key motivation for developing a good inference approximation here.

3 Expectation Propagation

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 [8, 11] to this problem, circumventing some caveats we have not seen being addressed before. EP computes a Gaussian approximation $Q(\boldsymbol{a})$ to the posterior $P(\boldsymbol{a}|D) \propto N(\boldsymbol{u}|\boldsymbol{X}\boldsymbol{a}, \sigma^2\boldsymbol{I})P(\boldsymbol{a})$. In order to motivate such, note that $\log P(\boldsymbol{a})$ is concave, since $\log P(a_i) = -\tau|a_i| + C$ is (see Eq. 2). Since the likelihood is a Gaussian function of $\boldsymbol{a}$, the log posterior $\log P(\boldsymbol{a}|D)$ is concave as well, thus unimodal, and a Gaussian approximation seems sensible. Other sparsity priors suggested in the context of robust regression, such as the Student-t or the "spike and slab" distribution[2], are not log-concave, and accurate approximate inference is very hard due to posterior multimodality.

If $P^{(0)} = P(\boldsymbol{u}|\boldsymbol{X}, \boldsymbol{a})$ is the Gaussian likelihood, the true posterior is $P(\boldsymbol{a}|D) \propto P^{(0)}(\boldsymbol{a}) \prod_i t_i(a_i)$, where $t_i(a_i) = (\tau/2)\exp(-\tau|a_i|)$. We refer to t_i as *sites*, and to $P^{(0)}$ as "base distribution", even though it cannot be normalized as probability distribution if $n > m$. An optimal Gaussian posterior approximation $Q(\boldsymbol{a})$ would be obtained by setting its mean and covariance to the true posterior statistics. This requires a n-dimensional non-Gaussian integration, which is intractable. However, we *are* able to do one-dimensional integrals involving a single site $t_i(a_i)$, and EP makes use of such iteratively in order to approximate the desired moments. The posterior approximation has the form $Q(\boldsymbol{a}) \propto P^{(0)}(\boldsymbol{a}) \prod_i \tilde{t}_i(a_i)$, where $\tilde{t}_i(a_i|b_i, \pi_i)$ are Gaussian functions, called *site approximations*. The $\boldsymbol{b}, \boldsymbol{\pi} \in \mathbb{R}^n$ are *site parameters* which are adjusted through EP. An EP update consists of computing the Gaussian *cavity distribution* $Q^{\backslash i} \propto Q\tilde{t}_i^{-1}$ and the non-Gaussian *tilted distribution* $\hat{P} \propto Q^{\backslash i} t_i$, then updating

[2] A mixture of a narrow and a wide Gaussian.

b_i, π_i such that the new Q' has the same mean and covariance as $\hat{P}$. This is iterated in some random ordering until convergence.

Denote the family of unnormalized Gaussians by

$$N^U(\boldsymbol{z}|\boldsymbol{r},\boldsymbol{R}) = \exp\left(-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{R}\boldsymbol{z} + \boldsymbol{r}^T\boldsymbol{z}\right), \qquad (3)$$

$\boldsymbol{R}$ being positive semidefinite. Note that some members are not probability distributions, in that they cannot be normalized. The family of proper Gaussian distributions is denoted by $N(\dots)$, and is a strict subset. Now, $P^{(0)}(\boldsymbol{a}) = N^U(\boldsymbol{a}|\boldsymbol{b}^{(0)}, \boldsymbol{\Pi}^{(0)})$ with $\boldsymbol{\Pi}^{(0)} = \sigma^{-2}\boldsymbol{X}^T\boldsymbol{X}$, $\boldsymbol{b}^{(0)} = \sigma^{-2}\boldsymbol{X}^T\boldsymbol{u}$. The site approximations[3] are $\tilde{t}_i(a_i) = N^U(a_i|b_i,\pi_i)$, so that Q is a Gaussian. One can show that as a consequence of the log concavity of t_i, we have that $\pi_i \geq 0$ for all i at all times.

In the particular case we are interested in, namely $n > m$, the standard application of EP fails, because the base distribution $P^{(0)}$ is not a proper Gaussian, it cannot be normalized and has infinite variance along almost all directions. We will refer to $P^{(0)}$ as "degenerate" in this case[4]. Furthermore, an efficient representation of Q is sought which scales with m, rather than with n.

3.1 Posterior Representation

In this Section, we present a representation of the posterior approximation $Q(\boldsymbol{a}) = N(\boldsymbol{h}, \boldsymbol{\Sigma})$, which allows efficient access to entries of $\boldsymbol{h}$, $\operatorname{diag}\boldsymbol{\Sigma}$ (marginal moments), and which can be updated robustly and efficiently for single site parameter changes. In the case $n > m$, Q is well-defined only if all $\pi_i > 0$, because $P^{(0)}$ cannot be normalized. For stability, we require that $\pi_i \geq \kappa$ at all times, where $\kappa > 0$ is a small constant. While EP is usually started from the base distribution, *i.e.* setting all site parameters to zero, we start with $\pi_i = \varepsilon > 0$, $b_i = 0$. Setting $\varepsilon = \tau^2/2$ ensures that $t_i(a_i)$ and $\tilde{t}_i(a_i)$ have the same variance (and mean).

Since $\pi_i > 0$, we can use the Sherman-Morrison-Woodbury formula [14] to write

$$\begin{aligned}\boldsymbol{\Sigma} &= \left(\sigma^{-2}\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\Pi}\right)^{-1} \\ &= \boldsymbol{\Pi}^{-1} - \boldsymbol{\Pi}^{-1}\boldsymbol{X}^T\left(\sigma^2\boldsymbol{I} + \boldsymbol{X}\boldsymbol{\Pi}^{-1}\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\boldsymbol{\Pi}^{-1},\end{aligned}$$

where $\boldsymbol{\Pi} = \operatorname{diag}\boldsymbol{\pi}$. We represent this by

$$\boldsymbol{L}\boldsymbol{L}^T = \sigma^2\boldsymbol{I} + \boldsymbol{X}\boldsymbol{\Pi}^{-1}\boldsymbol{X}^T,$$

where $\boldsymbol{L} \in \mathbb{R}^{m,m}$ is the lower triangular Cholesky factor with positive diagonal. Furthermore, let $\boldsymbol{\gamma} = \boldsymbol{L}^{-1}\boldsymbol{X}\boldsymbol{\Pi}^{-1}(\boldsymbol{b}^{(0)} + \boldsymbol{b})$, whence

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

thus both $\boldsymbol{h}$ and $\boldsymbol{\Sigma}$ are represented by $\boldsymbol{L}$, $\boldsymbol{\gamma}$.

For not too small κ (we used 10^{-8}), this representation is numerically stable. After an EP update $b_i \to b_i'$, $\pi_i \to \pi_i'$, the representation can be modified in $O(m^2)$, using a rank one Cholesky update of $\boldsymbol{L}$, $\boldsymbol{\gamma}$. Details are given in [16].

3.2 EP Updates

We motivated the EP update above as matching moments between a tilted and the new posterior distribution. For an update at site i, we require the marginal $Q(a_i) = N(h_i, \rho_i)$ only (details about EP in our notation can be found in [15]). If $\boldsymbol{v} = \boldsymbol{L}^{-1}\boldsymbol{X}^T_{\cdot,i}$, then $\rho_i = \pi_i^{-1}(1 - \pi_i^{-1}\|\boldsymbol{v}\|^2)$ and $h_i = \pi_i^{-1}(b_i^{(0)} + b_i - \boldsymbol{v}^T\boldsymbol{\gamma})$. Normally, cavity and tilted distribution would be computed from there in order to obtain the new π_i', b_i'. However, again the degenerate base distribution causes trouble. Recall from Section 3.1 that it is chiefly the positive π_i which keeps the variance of $Q(a_i)$ finite. Since $Q^{\backslash i}(a_i) \propto Q(a_i)\tilde{t}_i(a_i)^{-1}$, the cavity distribution is obtained by setting $\pi_i = 0$. The variance of $Q^{\backslash i}(a_i)$ is finite iff a_i is coupled to the other components of $\boldsymbol{a}$ through $P^{(0)}$. In our main application, however, this coupling is weak, resulting in $Q^{\backslash i}(a_i)$ having huge variance. Since the tilted distribution is $\hat{P}(a_i) \propto Q^{\backslash i}(a_i)t_i(a_i)$, we have to compute moments of the product of a very wide and a quite narrow function, which is numerically unstable. A way to circumvent this problem is to employ *fractional* EP updates (these have been introduced in a different context in [7]), which can be understood by imagining the site t_i being replaced by q fractional replicas $f_i = t_i^{1/q}$. The corresponding site approximation replicas $\tilde{f}_i = \tilde{t}_i^{1/q}$ have tied parameters. Namely, if $Q^{\backslash i} \propto Q\tilde{f}_i^{-1}$ and $\hat{P} \propto Q^{\backslash i}f_i$, we choose the new site parameters b_i', π_i' such that the moments of $\hat{P}$ and $\propto Q^{\backslash i}\tilde{f}_i'$ match. Now, the tilted distribution is obtained by dividing out a fraction of $\tilde{t}_i$ and multiplying in a fraction of t_i only, which alleviates the problem just mentioned. If $t_i(a_i) = \exp(-\tau|a_i|)$ (dropping the normalization), this is reduced to the standard EP update by substituting $\tilde{b}_i = b_i/q$, $\tilde{\pi}_i = \pi_i/q$, and $\tilde{\tau} = \tau/q$. For example,

[3] The fact that $\tilde{t}_i$ depends on a_i only, is a *consequence* of $t_i = t_i(a_i)$ and the way EP works, it is not a restricting assumption.

[4] Importantly, it is *not* the posterior over $\boldsymbol{A}$ matrices which is degenerate, and certainly the matrices $\boldsymbol{A}$ sampled from $Q(\boldsymbol{A})$ are square (by definition) and nonsingular almost surely, even for $m = 1$.

if $Q^{\setminus i}(a_i) = N(h_{\setminus i}, \rho_{\setminus i})$, then

$$\rho_{\setminus i} = \pi_i^{-1} q \left(\frac{q}{\pi_i^{-1}\|\boldsymbol{v}\|^2 + q - 1} - 1 \right),$$

$$h_{\setminus i} = \pi_i^{-1} q \left(\frac{b_i^{(0)} - \boldsymbol{v}^T \boldsymbol{\gamma}}{\pi_i^{-1}\|\boldsymbol{v}\|^2 + q - 1} + b_i/q \right).$$

Matching moments results in new values $\tilde{\pi}_i'$, $\tilde{b}_i'$. Rather than setting $\pi_i' = q\tilde{\pi}_i'$, we use a *damped* update, $\pi_i' = (1 - 1/q)\pi_i + \tilde{\pi}_i'$, $b_i' = (1 - 1/q)b_i + \tilde{b}_i'$, which has the desirable effect that $\hat{P}$ and the new Q' do have the same moments [7]. This can be understood by remembering the q replica interpretation of fractional updates: we move only by a fraction $1/q$ into the direction of new site parameters, accounting for the fact that we impose the change uniformly on all q replicas of $\tilde{f}_i$.

The moments of $\hat{P}(a_i)$ can be computed analytically, but it is fairly challenging to do this in a numerically stable manner. Some details are given in the appendix, and the complete derivation is given in [16]. Note that in the case of the sparse linear model with underdetermined likelihood ($m < n$), it is crucially important to compute EP updates very accurately, in spite of the fact that the posterior is log-concave (and unimodal)[5]. In applications where the Gaussian coupling potential $P^{(0)}$ is clearly non-degenerate, EP updates may be done to lesser accuracy, say by using Gaussian quadrature.

Finally, our experimental design application to gene network identification requires updating the posterior factors $Q(\boldsymbol{a}^{(j)})$ in a sequential manner, including new observations $(\boldsymbol{x}_*, \boldsymbol{u}_*)$ one at a time. For a single factor, we first update the base distribution $P^{(0)}$ through a rank one Cholesky update of the representation, costing $O(m^2)$ [16]. This followed by updating the site parameters $\boldsymbol{b}$, $\boldsymbol{\pi}$ through an EP sweep over all sites.

4 Sequential Optimal Design

The role of sequential optimal design for saving on expensive experiments has already been motivated in Section 2. In the sparse linear model, the general design problem can be formulated as follows. A complete observation is given by $(\boldsymbol{x}_*, u_*)$, and a *candidate* can be seen as incomplete observation. For example, in the standard design setup, a candidate is given by $\boldsymbol{x}_*$, with u_* unknown, while in our setup of interest, u_* is given and $\boldsymbol{x}_*$ unknown. Now, given a set of candidates, the design problem requires to decide which of these should be *queried* next, meaning that a value for the unknown part is sampled from the true underlying distribution (sometimes called the "oracle"). The goal here is to obtain as much new information about the unknown $\boldsymbol{a}$ as possible. Assuming (for the moment) that $(\boldsymbol{x}_*, u_*)$ is completely known for a candidate, natural design scores quantify the decrease in posterior uncertainty or gain in information from the current posterior Q to the novel Q',obtained by including $(\boldsymbol{x}_*, u_*)$ into the data D. In this paper, we concentrate on the information gain[6] $\mathrm{D}[Q' \,\|\, Q] = \mathrm{E}_{Q'}[\log Q' - \log Q]$. A large information gain means that Q' is different from Q, thus much novel information is gained from $(\boldsymbol{x}_*, u_*)$. Now, $(\boldsymbol{x}_*, u_*)$ is *not* fully known for any candidate. Bayesian methodology dictates that any uncertainty in $(\boldsymbol{x}_*, u_*)$ is averaged over its current predictive distribution. For example, in the standard design setup, where $\boldsymbol{x}_*$ is given, we would use $Q(u_*|\boldsymbol{x}_*, D) = \int P(u_*|\boldsymbol{x}_*, \boldsymbol{a}) Q(\boldsymbol{a}|D)\, d\boldsymbol{a}$, which is Gaussian, and a score for $\boldsymbol{x}_*$ would be the expected information gain $\mathrm{E}_{Q(u_*|\boldsymbol{x}_*, D)}[\mathrm{D}[Q' \,\|\, Q]]$, which could easily be approximated using Gaussian quadrature.

Now, optimal design for the gene network application of Section 2 differs from the standard design setup, in that a candidate is given by $\boldsymbol{u}_*$, and $\boldsymbol{x}_*$ is unknown. However, we can still use the same argumentation to arrive at a design score. First, $(\boldsymbol{x}_*, \boldsymbol{u}_*)$ renders information for n independent posterior factors $Q(\boldsymbol{a}^{(j)})$, thus the (complete) information gain score is the sum of $\mathrm{D}[Q' \,\|\, Q]$ over these factors, where $(\boldsymbol{x}_*, u_{*,j})$ is included for the j-th factor. Next, the predictive distribution $Q(\boldsymbol{x}_*|\boldsymbol{u}_*, D) = \int P(\boldsymbol{x}_*|\boldsymbol{u}_*, \boldsymbol{A}) Q(\boldsymbol{A}|D)\, d\boldsymbol{A}$ is not a simple distribution (like a Gaussian), but we can easily sample from it by first drawing $\boldsymbol{A} \sim Q(\boldsymbol{A}|D)$, then[7] $\boldsymbol{x}_* = \boldsymbol{A}^{-1}(\boldsymbol{u}_* - \boldsymbol{\varepsilon})$, $\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Our *information gain* score in the gene network application is

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

where the expectation is approximated by using a number of independent samples $\boldsymbol{x}_*$.

Note that for fixed $(\boldsymbol{x}_*, u_*)$, Q' is obtained from Q by first modifying the base distribution $P^{(0)}$ corresponding to the inclusion, then updating the site parameters $\boldsymbol{b}$, $\boldsymbol{\pi}$. The latter requires EP updates until new convergence is established, which is fairly expensive and may prevent us scoring many candidate pairs. A simpler alternative is to approximate $\mathrm{D}[Q' \,\|\, Q]$ by modifying $P^{(0)}$ only, but keeping the old site parameters, when defining Q' for scoring. In this paper, we concentrate on this simpler alternative. Details for the full infor-

[5] A reason may be that the Laplace density is "just about log-concave": $\propto \exp(-|\cdot|^\alpha)$ is *not* for any $\alpha < 1$.

[6] We also did experiments with the entropy reduction $\mathrm{E}_{Q'}[\log Q'] - \mathrm{E}_Q[\log Q]$, which did not lead to significantly different results.

[7] We use a LU decomposition of $\boldsymbol{A}$. The same set of decomposed $\boldsymbol{A}$'s can be used to score *all* candidates.

mation gain criterion are given in [16]. Note that the latter score is much harder to compute in a stable manner, and it did not result in significant improvements in preliminary experiments. The relative entropy between Gaussians is well known, and we exploit the fact that $(\boldsymbol{\Sigma}')^{-1} = \boldsymbol{\Sigma}^{-1} + \sigma^{-2}\boldsymbol{x}_*\boldsymbol{x}_*^T$, using the Sherman-Morrison-Woodbury formula in order to compute the score in $O(m^2)$. Details can be found in [16].

5 The Marginal Likelihood

Apart from $\boldsymbol{a}$, the linear model comes with additional *hyperparameters*, namely σ^2, τ, and free parameters in $\boldsymbol{X}$. A powerful empirical Bayesian way of estimating such hyperparameters works by maximizing the marginal likelihood

$$P(D) = \int \prod_{i=1}^{n} t_i(a_i) P^{(0)}(\boldsymbol{a})\, d\boldsymbol{a}.$$

The computation of $P(D)$ is intractable for the sparse linear model, but an approximation of it may be obtained within the EP framework. While we do not use the marginal likelihood in our experiments here, we provide a derivation for future use.

Recall that EP works by matching moments of first and second order between marginals $Q(a_i)$ and tilted distributions $\hat{P}(a_i)$. In order to approximate the normalization constant $P(D)$, we match zero-order moments as well. To this end, we replace $t_i(a_i)$ by $C_i\tilde{t}_i(a_i)$, $\tilde{t}_i(a_i) = N^U(a_i|b_i,\pi_i)$, and we set the C_i such that $Q(a_i)$ and $\hat{P}(a_i)$ have the same normalization constant as well: $\log C_i = \log Z_i - \log \tilde{Z}_i$, where $Z_i = \mathrm{E}_{Q^{\setminus i}}[t_i(a_i)]$, $\tilde{Z}_i = \mathrm{E}_{Q^{\setminus i}}[\tilde{t}_i(a_i)]$. This can be computed in a final sweep, once EP has converged. Plugging in $C_i\tilde{t}_i(a_i)$ for $t_i(a_i)$ results in the following approximation of $\log P(D)$:

$$L = \sum_{i=1}^{n} \log C_i + \Phi[Q] - \frac{1}{2\sigma^2}\|\boldsymbol{u}\|^2$$

where $\Phi[N(\boldsymbol{h},\boldsymbol{\Sigma})] = (1/2)\log|2\pi\boldsymbol{\Sigma}| + (1/2)\boldsymbol{h}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{h}$ is the log partition function of a Gaussian. Furthermore, if $\boldsymbol{\theta}^{(0)}$ are parameters of $P^{(0)}$, then $\nabla_{\boldsymbol{\theta}^{(0)}} L = \mathrm{E}_Q[\nabla_{\boldsymbol{\theta}^{(0)}} \log P^{(0)}(\boldsymbol{a})]$. It is important to note that once approximate inference has been done to obtain Q, the gradient computation scales linearly in the number of hyperparameters, thus L can in principle be optimized over many of these. A derivation of these facts is given in [15]. A concrete expression for L and its gradient are given in [16].

Note that the computation of L and its gradient requires doing full (non-fractional) EP updates until convergence, because the derivation of either uses the fact that an EP fixed point has been reached. Recall from Section 3.2 that in the presence of a degenerate $P^{(0)}$, fractional updates may be required for stable computation. If $P^{(0)}$ is strongly degenerate, such as in the regulatory networks setup with few experiments done ($m \ll n$), full EP updates cannot be done properly even at the fractional stationary point. In these cases, L cannot be computed. Finding an approximation of $\log P(D)$ for the fractional case is subject to future work. For other $P^{(0)}$, full EP updates can be done after a few initial fractional ones. Preliminary experiments indicate that an application to sparse coding of natural images (see Section 7) is of this kind.

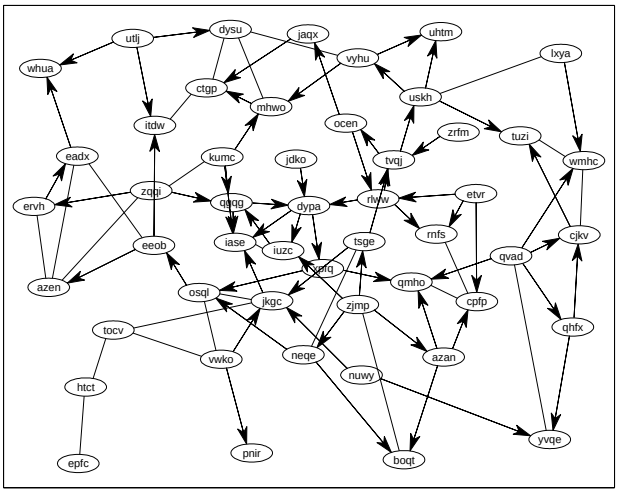

Figure 1: *An example network with random gene names. Undirected edges in the plot correspond to a bidirectional relationship in the true network.*

6 Experiments

The application of the sparse linear model to active regulatory network recovery was motivated in Section 2. Here, we describe some preliminary experiments. All our results are averaged over 100 runs. In one run, we first generate a graph and matrix $\boldsymbol{A}$ as follows. Biological networks are found to be sparsely connected and showing small-world properties [20]. We sample such networks with $n = 50$ nodes, using an algorithm described in [1], the average number of parents being 2.4. An example network is shown in Figure 1. Given the graph, the non-zero values $a_{i,j} \sim U[-1,1]$ independently. Given $\boldsymbol{A}$, we sample 1000 candidate controls $\boldsymbol{u}_*$ and corresponding experimental outcomes $\boldsymbol{x}_*$. Since up-/downregulating a specific gene is an expensive procedure, we focus on sparse controls $\boldsymbol{u}_*$ for biological relevance, setting 3 randomly selected entries to $\pm\frac{1}{\sqrt{3}}$ (at random), so that $\boldsymbol{u}_*$ has norm one.

$\boldsymbol{x}_*$ is sampled from the model[8], where the noise standard deviation is $\sigma = 0.01$. In general, σ^2 has to be determined from biological prior knowledge or experiments on related systems. Here, we consider σ to be known.

Several methods, to be described shortly, are now compared on this pool of $(\boldsymbol{x}_*, \boldsymbol{u}_*)$ pairs. A method has access to all $\boldsymbol{u}_*$ values, and may sequentially request $\boldsymbol{x}_*$ for some $\boldsymbol{u}_*$, one in each iteration. Each method is tested after each iteration, by comparing its prediction against the true $\boldsymbol{A}$. We allow for 50 iterations, as many as there are nodes, after which we would expect a well-performing method to reconstruct the graph fairly reliably. Recall that we have a continuous posterior over $\boldsymbol{A}$, from which we need to output a prediction of the graph. For some small ρ (we use $\rho = 0.1$; while the prediction depends on ρ, it is stable except for extreme choices), we predict an edge (i, j) iff $Q(\{|a_{i,j}| > \rho\})$ is larger than some threshold. By varying the threshold, we can perform a standard ROC analysis. Since networks of interest are sparse, the number of false positives is potentially large, which is why only the leftmost part of the ROC curve is interesting. The evaluation score we use, called iAUC, is the area under the ROC curve up to a number of false positives equal to the true number of edges, normalized to lie in $[0, 1]$. The hyperparameter τ was set using the following heuristic: under the Laplace prior, the expected number of parents predicted per gene is $n \exp(-\tau\rho)$. Given the true average number of parents $\bar{d}$, we set $\tau = -\rho^{-1} \log(\bar{d}/n)$. In our experiments, $\bar{d}$ is known. In practice, it could be estimated from networks of similar type[9].

The methods we consider here all use (approximate) Bayesian inference in the linear model. We compare using Laplace against Gaussian priors on $\boldsymbol{A}$, and random inclusions versus active design choices. For random inclusions, the pool of candidates $\boldsymbol{u}_*$ is sampled without replacement, and active design is done using the information gain score. When using a Gaussian prior, we choose its variance such that $P(\{|a_{i,j}| > \rho\})$ is the same as for the Laplace prior, so that the difference is merely in the shape of the priors. Results are shown in Figure 2.

From these results, it is clear that using a Laplace prior instead of a Gaussian one pays off significantly, even if only random inclusions are done. This means that the sparse linear model is clearly superior to the standard normal linear model, even in situations where experiments are not actively designed. Both methods with Gaussian prior do not attain maximum iAUC scores (meaning reliable recostruction of the whole network) even with 50 experiments. We should note that dealing with a Laplace prior is more costly than using the standard Gaussian one, but only by a small constant factor in our experience so far. One run with "Laplace Design" (50 inclusions, 1000 candidates, 20 samples of $\boldsymbol{A}$ per information gain score) takes about 6 minutes on a Desktop PC (2 GHz Pentium III).

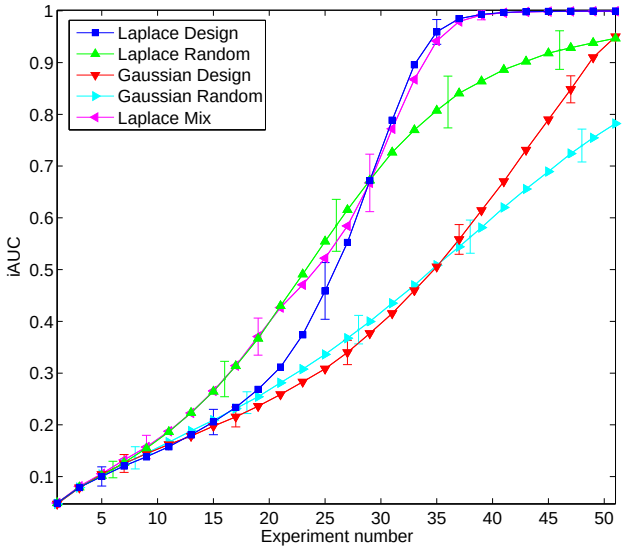

Figure 2: *Network recovery using different methods: Laplace versus Gaussian prior, optimal design versus random selection. Mixed method: first 20 controls drawn randomly, optimal design for the rest.*

Comparing random versus designed inclusions, it is also clear in both cases (Laplace and Gaussian) that the latter performs much better. In the Laplace case, the whole network is reliably recovered after about 35 experiments with designed inclusions, while random inclusions only attain $iAUC = 0.9$ after all 50 inclusions. An interesting effect, pronounced more in the Laplace case, is that active design seems to hurt initially, until about $25 = 50/2$ experiments have been done. While we do not have a conclusive explanation for this observation, we note that active design on the basis of a very uncertain posterior is considered unreliable in general. It is encouraging though that the active Laplace variant steeply recovers beyond 25 inclusions and is not harmed by the initial underperformance. We also tested a mixed variant, called "Laplace Mix" in Figure 2, for which the first 20 inclusions were selected at random, the remaining ones were chosen actively. There is no significant under-

[8]More realistic experiments, where $\boldsymbol{x}_*$ is obtained by forward simulation of a biologically motivated nonlinear ODE system, have been obtained and been submitted for publication.

[9]Our method is fairly robust to different τ around this estimate, but the number of required experiments increases sharply if τ is chosen far from the heuristic estimate.

performance anywhere for this method.

Note that in the experimental design setup we considered here, n inclusions have to be decided upon. After each inclusion, we have to update n posterior factors over n variables each, and an EP sweep for one of these factors costs $O(n^3)$ during later stages. These update sweeps clearly dominate the computational effort. Thus, a straightforward implementation (as used here) requires $O(n^5)$ time. We also need n posterior representations, so $O(n^3)$ memory. For large n (say beyond 400), a more careful implementation may be necessary. First, we can parallelize the posterior representations and EP update computations up to a factor of n. Next, doing EP updates for all sites after each inclusion is very conservative. An immediate idea would be to identify, for each factor i separately, those sites j whose marginals $Q(a_j)$ change most upon the inclusion of a point $(\boldsymbol{x}_*, u_{*,i})$ into the likelihood part $P^{(0)}$ (but *before* any EP updates), then do EP updates for these sites only. This would cut down the computational cost by a factor up to n. The LU decompositions of sampled $\boldsymbol{A}$ (required to compute the design scores, see Section 4) may also become problematic, although they only contribute $O(n^3)$ for each inclusion. We would recommend to sparsify $\boldsymbol{A}$ in this case, and use sparse decomposition code.

7 Discussion

We have shown how to perform accurate approximate Bayesian inference in the linear model with a Laplace prior very efficiently, and how this can be used to address tasks such as optimal design and hyperparameter estimation. These capabilities were demonstrated on an application to identification of gene regulatory networks.

The idea of L_1 regularization has been used in very many contexts. The maximum a posteriori (MAP) treatment of the sparse linear model has been proposed as *Lasso* [18] and as *basis pursuit* [2]. The linear model can be configured with other sparsity-inducing priors, in order to obtain robust variants of linear regression. The prime advantage of an MAP treatment is that the fitting to data can be done very efficiently. On the other hand, MAP as an approximation to Bayesian inference is poor in this case. We have demonstrated a few advantages of going the full Bayesian way 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 in [13], but they do not address the problem of optimal design.

An approximate Bayesian method for the linear model with Student-t prior has been given in [19]. In their case, the posterior is not log-concave and multimodal[10]. Furthermore, their family in which Gaussian posterior approximations live, has less variational parameters than the one used by EP, which may lead to less accurate approximations. A Markov chain Monte Carlo treatment of the sparse linear model is proposed in [12], where the Laplace distribution is written as scale mixture of Gaussians, and a block Gibbs sampler is developed. While this approach has the potential of being exact in the limit of large running time[11], it is still much slower than our approximate method and may not be applicable to many large tasks addressed in Machine Learning. Furthermore, the marginal likelihood may not be obtained directly from their method. Comparing our approach to these approximate Bayesian alternatives, none of which consider experimental design, is subject to future work.

More realistic experiments for gene network recovery are in preparation, using ODE generators in order to simulate from a network. In this context, more elaborate noise modeling, dynamic aspects, and other realistic types of external control will be looked into.

We also plan to apply our method to the problem of learning and analyzing image codes [10, 5], with the aim of understanding properties of visual neurons in the brain. In this context, the sparse linear model has been proposed as a realistic model, in which codes can be learned by maximizing the marginal likelihood. The marginal likelihood approximation of Section 5 is potentially more accurate than the one used in [5], and it will be interesting to test their hypothesis using our framework.

The Bayesian sparse linear model may have many other applications, given that its MAP variants (Lasso, basis pursuit) are very widely used. For example, EP has been applied to approximate inference in generalized linear models (GLMs) with a Gaussian prior, while our application here is to a linear model with a Laplace prior. An application to a GLM with a Laplace prior is subject to future work.

8 Appendix

The EP update for Laplace sites $t_i(a) = \exp(-\tau|a|)$, $\tau > 0$ can be done analytically, but the numerically stable computation is fairly involved. We present the main points of our approach here, a complete exposition is given in [16]. We need the

[10] Their approach can be applied to the linear model with Laplace prior, by using the scale mixture decomposition given in [12].

[11] The log-concavity of the posterior should translate into fairly fast mixing.

moments $I_k = \mathrm{E}_{N(h,\rho)}[a^k t_i(a)]$, $k = 0, 1, 2$, where $a = a_i$, $h = h_{\setminus i}$, $\rho = \rho_{\setminus i}$. Assume for now that $\tau = 1$. Then, $I_0 = \tilde{I}_0(h) + \tilde{I}_0(-h)$, where

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

Here, Φ is the cumulative distribution function of $N(0, 1)$. Now, we easily see that $\tilde{I}_0(|h|) \geq \tilde{I}_0(-|h|)$, so that

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

permits a stable computation. We now use the well known asymptotic expansion

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

If $F(x) = \log(1 - \Phi(x))$, we use this expansion up to $1 - 7x^{-2}$ for $x > 5$, but compute $F(x)$ exactly otherwise[12]. We have that $\log \tilde{I}_0(|h|) = \rho/2 - |h| + F(\rho^{1/2} - |h|\rho^{-1/2})$, and

$$\begin{aligned}R := \frac{\tilde{I}_0(-|h|)}{\tilde{I}_0(|h|)} = \exp\Big(&2|h| + F(\rho^{1/2} + |h|\rho^{-1/2}) \\ &- F(\rho^{1/2} - |h|\rho^{-1/2})\Big),\end{aligned}$$

which gives $\log I_0$. The computation of I_1, I_2 uses the same ideas, it is given in [16]. The mean of $\hat{P}(a)$ is $\hat{h} = I_1/I_0$, the variance is $\hat{\rho}^2 = I_2/I_0 - \hat{h}^2$. If $\tau \neq 1$, we simply plug in $h = \tau h_{\setminus i}$, $\rho = \tau^2 \rho_{\setminus i}$ above, and multiply $\hat{h}$ by τ, $\hat{\rho}$ by τ^2. Finally, a fractional EP update is done by simply scaling τ accordingly.

Acknowledgments

Supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, 2002.

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

[3] 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.

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

[5] 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.

[6] D. MacKay. Information-based objective functions for active data selection. *N. Comp.*, 4(4):589–603, 1991.

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

[8] Thomas Minka. Expectation propagation for approximate Bayesian inference. In *Uncertainty in AI 17*, 2001.

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

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

[11] Manfred Opper and Ole Winther. Gaussian processes for classification: Mean field algorithms. *N. Comp.*, 12(11):2655–2684, 2000.

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

[13] R. Peeters and R. Westra. On the identification of sparse gene regulatory networks. In *Proc. 16th Intern. Symp. on Mathematical Theory of Networks (MTNS 2004)*, 2004.

[14] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

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

[16] M. Seeger, F. Steinke, and K. Tsuda. Bayesian inference and optimal design in the sparse linear model. Technical report, Max Planck Institute for Biologic Cybernetics, Tübingen, Germany, 2006. See `www.kyb.tuebingen.mpg.de/bs/people/seeger`.

[17] 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.

[18] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Roy. Stat. Soc. B*, 58:267–288, 1996.

[19] Michael Tipping. Sparse Bayesian learning and the relevance vector machine. *J. M. Learn. Res.*, 1:211–244, 2001.

[20] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440, 1998.

[12] Note that the simpler approximation $1 - \Phi(x) \approx N(x)/x$ is insufficient and leads to complete failure of EP on most tasks.