Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. TR-175

Large Scale Variational Inference and Experimental Design for Sparse Generalized Linear Models

Matthias W. Seeger,[1] Hannes Nickisch,[1]

September 2008

[1] Department Schölkopf. Contact: seeger@tuebingen.mpg.de

Dagstuhl Seminar Proceedings 09181
Sampling-based Optimization in the Presence of Uncertainty
http://drops.dagstuhl.de/opus/volltexte/2009/2114

Large Scale Variational Inference and Experimental Design for Sparse Generalized Linear Models

Matthias W. Seeger, Hannes Nickisch

Abstract. Sparsity is a fundamental concept of modern statistics, and often the only general principle available at the moment to address novel learning applications with many more variables than observations. While much progress has been made recently in the theoretical understanding and algorithmics of sparse point estimation, higher-order problems such as covariance estimation or optimal data acquisition are seldomly addressed for sparsity-favouring models, and there are virtually no algorithms for large scale applications of these. We provide novel approximate Bayesian inference algorithms for sparse generalized linear models, that can be used with hundred thousands of variables, and run orders of magnitude faster than previous algorithms in domains where either apply. By analyzing our methods and establishing some novel convexity results, we settle a long-standing open question about variational Bayesian inference for continuous variable models: the Gaussian lower bound relaxation, which has been used previously for a range of models, is proved to be a convex optimization problem, if and only if the posterior mode is found by convex programming. Our algorithms reduce to the same computational primitives than commonly used sparse estimation methods do, but require Gaussian marginal variance estimation as well. We show how the Lanczos algorithm from numerical mathematics can be employed to compute the latter.

We are interested in Bayesian experimental design here (which is mainly driven by efficient approximate inference), a powerful framework for optimizing measurement architectures of complex signals, such as natural images. Designs optimized by our Bayesian framework strongly outperform choices advocated by compressed sensing theory, and with our novel algorithms, we can scale it up to full-size images. Immediate applications of our method lie in digital photography and medical imaging.

We have applied our framework to problems of magnetic resonance imaging design and reconstruction, and part of this work appeared at a conference (Seeger et al., 2008). The present paper describes our methods in much greater generality, and most of the theory is novel. Experiments and evaluations will be given in a later paper.

---

1 Introduction

Generalized linear models are cornerstones of applied statistics, and are also very frequently used in machine learning. In many applications from low level computer vision, bio-informatics, neuroscience, information retrieval, adaptive filtering and control, or medical image reconstruction, a vast number of features could potentially be used, although the important ones for any given task are not known *a priori*. In such contexts, the concept of *sparsity regularization* or *sparsity priors* is of central importance, either to select a relevant subset of features in a data-driven manner, or to improve estimation or inference by conditioning them on the assumption that a solution free of noise ought to be sparsely distributed.

Sparsity is a fundamental concept of modern, high-dimensional statistics. For signal classes such as natural images, it can be motivated by direct empirical analysis: images *are* sparsely distributed in certain transform domains. In general, it complements linearity as common model assumption in a way which is strictly different from the traditional least squares (LS) idea. If a signal is represented as linear combination over some basis set (or dictionary), we may enforce "simplicity" of this representation in different ways. LS, or Gaussian, regularization penalizes the size of all linear coefficients uniformly, so that simple signals have smallish coefficients throughout. In contrast, sparsity regularization enforces most coefficients to be tiny or even zero, at the expense of allowing for a few large ones, so that simple signals are dominantly explained by few basis functions. The rather general success of sparsity statistics at denoising signals comes from the fact that, due to the central limit theorem, noise is typically Gaussian (coming from many small, independent error sources), while real-world signals are typically

highly structured, therefore rather sparsely distributed. More details about the statistical role of sparsity can be found in (Seeger, 2008).

When sparsity is enforced in LS estimation or approximate Bayesian inference, the method is to concentrate on a small subset of explanatory variables, about which nothing is known explicitly beforehand, and one might guess that a combinatorial problem is lurking behind the scenes. Indeed, a harsh formulation of sparse estimation, known as L_0 regularization (see Section 5.2), is NP complete. Fortunately, a surge of recent activity has established that in many practically relevant cases, feature selection or sparse estimation can be performed by *convex programs* (Donoho, 2006; Candès et al., 2006; Donoho and Elad, 2003), which can be solved very efficiently. Modern algorithms achieve scalability to very many variables by reducing their dominating efforts to reweighted least squares problems, for which efficient code is in common use.

In this paper, we are interested in Bayesian inference and applications thereof, problems which are distinctly different from sparse estimation. For example, if $\boldsymbol{u}$ is an unknown image, and $\boldsymbol{y}$ are linear measurements thereof, sparse estimation is useful in order to reconstruct $\boldsymbol{u}$ from $\boldsymbol{y}$ through a single point estimate. Traditional LS estimation is equivalent to maximum-likelihood estimation under the assumption of Gaussian noise. In sparse estimation, we regularize the LS estimate by minimizing the sum of the negative log likelihood $-\log P(\boldsymbol{y}|\boldsymbol{u})$ and a sparsity-enforcing regularization term acting on $\boldsymbol{u}$. From a statistical viewpoint, a single point estimate is unsatisfying, since it does not come with an assessment of uncertainty. For example, if we had an additional measurement y_i on top of $\boldsymbol{y}$, would our estimate improve, and if so, how exactly? In Bayesian statistics, the *posterior distribution* is a complete representation of the uncertainty in a reconstruction of $\boldsymbol{u}$ from data $\boldsymbol{y}$. The sparsity regularization term is replaced by a sparsity prior $P(\boldsymbol{u})$, a distribution over images $\boldsymbol{u}$, and the posterior is obtained by Bayes formula as $P(\boldsymbol{u}|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{u})P(\boldsymbol{u})$. A sparse estimator is recovered as the posterior mode: $\hat{\boldsymbol{u}} = \mathrm{argmax}_{\boldsymbol{u}}\, P(\boldsymbol{u}|\boldsymbol{y}) = \mathrm{argmin}_{\boldsymbol{u}}[-\log P(\boldsymbol{y}|\boldsymbol{u}) - \log P(\boldsymbol{u})]$, the *maximum a-posteriori* (MAP) estimator. Here, $-\log P(\boldsymbol{u})$ plays the role of the sparsity regularizer. But the posterior $P(\boldsymbol{u}|\boldsymbol{y})$ is much more than just its mode, and can be used for tasks beyond point estimation. In this paper, we will demonstrate a particular application, which is driven by posterior covariances, and for which image point estimation would be of no value at all. While many of the recently studied sparse estimation settings correspond to (convex) posterior *maximization* in sparse linear models, real Bayesian applications require us to *integrate* over the posterior, a computation which is less well understood, and is thought to be much harder to do in general.

Nevertheless, our motivation is to apply Bayesian inference in domains where sparse estimation is typically used. This means that our algorithms have to run efficiently for very large numbers of variables, yet still produce useful uncertainties along with point estimates. We focus on a variational relaxation of inference here, which has been used before on problems of moderate size. The range of models in the scope of our results here is not restricted to Gaussian-linear likelihoods, but contains generalized linear models as well, such as logistic regression. We establish that a wide range of instances of these variational relaxations are *convex minimization* problems with unique solutions, an insight which is novel to the best of our knowledge. For example, variational Bayesian treatments of sparsity models with Laplace sites, of binary classification models, or of combinations thereof, are shown to be convex problems. In a nutshell, whenever potentials in the model are amenable to Fenchel duality lower bounding (Jaakkola, 1997; Palmer et al., 2006), and the log posterior is concave (meaning that MAP estimation is convex), the variational relaxation is proved to be convex.

Beyond this insight, which strengthens a range of prior work, we present novel *scalable* algorithms to solve this variational relaxation of Bayesian inference for sparse (generalized) linear models very efficiently, whether it is convex or not.[1] Just as for convex estimation, scalability means that all major computations are reduced to standard primitives of large scale numerical mathematics and least squares estimation, which have received much attention already in most computational application fields. This has the very considerable advantage that highly optimized code can be made to use, and that our methods can be imported into many large scale applications with minor efforts. We reduce inference to a sequence of reweighted least squares problems, as well as Gaussian marginal variance computations. These can be reduced further (by standard numerical mathematics algorithms) to *matrix-vector multiplication* (MVM) primitives with model matrices, any structure exploitation of which has a direct impact on our dominating computations, with no further heuristics to be tuned.

Our special interest is in Bayesian experimental design, which is a framework for improving measurement architectures automatically, with the aim of obtaining reconstructions of equivalent quality under lower cost. In the image reconstruction example, which measurement design allows for the best sparse reconstruction of $\boldsymbol{u}$ from $\boldsymbol{y}$? To answer this question, we compute design scores (expectations over the current posterior), whose inspection

[1] A local minimum is found if the relaxation is not convex.

reveals directions of improvement for the current design. The power of our approach in practice has been demonstrated for a number of applications already (Steinke et al., 2007; Seeger, 2008; Seeger and Nickisch, 2008), but the variational approximations and algorithms used there are not scalable and cannot be used in the large scale domains of interest here. With the novel methods presented here, these settings can be lifted to full-size images, and problems in medical image reconstruction can be addressed.

On a high level, our approach can be understood as a relaxation of Bayesian inference for distinctly non-Gaussian generalized linear models to a (small) sequence of Gaussian linear model computations, such as computing means and marginal variances. The considerable experience with Gaussian random fields in (say) low-level computer vision can therefore be used to address inference in non-Gaussian models, which represent natural image characteristics such as sparsity much better than Gaussian models do, or in models for discrete observations.

Beyond pure Bayesian applications, the problem of finding very good or even optimal designs for subsequent *sparse* image reconstruction does not have a satisfying solution yet. While much is known about good measurements supporting linear LS estimation, nonlinear sparse reconstruction corrects for many shortcomings of the latter, so that the relevance of many linear design properties is most probably diminished. Recent theoretical results about sparse convex estimation (Donoho, 2006; Candès et al., 2006; Donoho and Elad, 2003) are not helpful in that respect, since they focus on truly sparse signals $\boldsymbol{u}$, while natural images are not well described by sparsity alone (Weiss et al., 2007). The inappropriateness of theoretical properties such as maximal incoherence or RIP as design principles for natural image reconstruction has been demonstrated in (Seeger and Nickisch, 2008). However, we observe in practice that designs optimized by our approach support subsequent sparse MAP reconstruction successfully, indeed as well or better than other "more Bayesian" estimates linked to the posterior, such as its mean. Our method can be used for design optimization, if the objective is sparse MAP estimation. It solves the problem of "learning compressed sensing" (Weiss et al., 2007) for large scale signals. Moreover, since current scalable MAP estimation codes, or even LS estimation codes, are based on much the same underlying primitives, the added effort of setting up our algorithms is minor.

The variational relaxation of inference employed here is not novel (Girolami, 2001; Palmer et al., 2006), but previously known algorithms for solving it are orders of magnitude slower than our approach on the problems considered here. Moreover, their convergence behaviour is not well characterized. Our contribution basically settles the question under which conditions the general approach of (Palmer et al., 2006) leads to a convex optimization problem. Our algorithm development owes ideas to (Wipf and Nagarajan, 2008), whose interest is in aggressively sparse estimation beyond convex MAP. Our framework moves considerably beyond their method, in generality, scope, and practical realization. Our convexity proof is novel, and our methods are applicable to more general models. Our proofs are based on convexity results for certain log determinants, which are novel in machine learning to our knowledge, and may have other applications there. Our interest is in estimating Bayesian uncertainties, information which is essentially destroyed in successful sparse estimation, as will be discussed in some detail. The major computational benefit of sparse estimation, namely many variables becoming exactly equal to zero rapidly, is responsible for this information loss, so that successful sparse Bayesian inference has to be implemented efficiently *without relying on exact sparsification*.

The structure of the paper is as follows. The sparse linear model is introduced in Section 2, where we also discuss our variational relaxation of Bayesian inference. In Section 3, we prove novel convexity properties of this relaxation. Our scalable algorithms for solving the variational problem are introduced in Section 4. In Section 5, we present some extensions, and discuss the relationship to MAP estimation and algorithmic aspects of sparsity. Bayesian sequential experimental design is discussed in Section 6. We close with a discussion, putting our work into context, and suggesting applications which would directly benefit from it. Experimental results for our novel methods will be presented in a later paper.

2 Sparse Bayesian Inference. Variational Approximations

In this section, we introduce sparse linear models, which can be used to represent image reconstructions together with remaining uncertainties, as noted in Section 1. Bayesian posterior (or inference) computations, which are required to compute queries for optimizing the measurement designs, cannot be done exactly for these models. We introduce a general variational inference approximation, based on the idea of convex inequalities for the cumulant generating function of the posterior.

Consider the problem of (natural) image reconstruction introduced in Section 1, which has real-world applications in computational photography and medical imaging (for example, magnetic resonance imaging or positron-emission tomography). The image $\boldsymbol{u} \in \mathbb{R}^n$ of n pixels has to be estimated from linear measurements $\boldsymbol{y} \in \mathbb{R}^m$,

where $m \ll n$ in many situations of practical interest. Such measurements suggest a *linear model*

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

where $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ is the *design matrix*, and $\boldsymbol{\varepsilon}$ is Gaussian noise of variance σ^2. The implied likelihood is Gaussian: $P(\boldsymbol{y}|\boldsymbol{u}) = N(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{u}, \sigma^2\boldsymbol{I})$, and maximum-likelihood estimation leads to the famous normal equations of linear least squares (LS) estimation. However, for $m \ll n$, these methods can work poorly for image reconstruction, mostly because they do not reflect image properties of $\boldsymbol{u}$ at all. For example, it is an established fact of nature that the projections $\boldsymbol{p}_j^T\boldsymbol{u}$ of a natural image under zero-mean filters $\boldsymbol{p}_j$, such as nearest neighbour differences (image gradient), wavelet or Fourier coefficients, are distibuted in a distictly non-Gaussian way: most coefficients are close to zero, while a certain fraction have significant sizes (Simoncelli, 1999). If such *super-Gaussian* properties (also known as *sparsity* of natural images) are encoded in a prior distribution $P(\boldsymbol{u})$, reconstructions are typically improved. The non-Gaussianity of $P(\boldsymbol{u})$ is important here. While a Gaussian prior leads to much simpler, analytically tractable computations, the main improvement comes from non-Gaussian properties. More details about this point can be found in (Seeger, 2008).

In this paper, we concentrate on priors of the form $P(\boldsymbol{u}) \propto \prod_{i=1}^q t_i(s_i)$, where $\boldsymbol{s} = \boldsymbol{B}\boldsymbol{u}$. In the image applications we are interested in here, $\boldsymbol{B}$ may contain local derivative, wavelet, or Fourier filters, but many other applications come with this structure. For example, with time series data, $\boldsymbol{B}$ may encode temporal differences. The matrices $\boldsymbol{X}$ and $\boldsymbol{B}$ model the couplings of variables. On the lowest level, exploitable structure in these is what renders our algorithms scalable. An example for sparsity-favouring sites t_i are *Laplace* (or *double exponential*) potentials

$$t_i(s_i) = \frac{\tilde{\tau}_i}{2} e^{-\tilde{\tau}_i|s_i|}, \quad \tilde{\tau}_i = \tau_i/\sigma > 0. \tag{2}$$

For this particular prior and the Gaussian likelihood (1), the MAP estimator is a (convex) quadratic program, the special case $\boldsymbol{B} = \boldsymbol{I}$ is known as *Lasso* (Tibshirani, 1996). In general, if all $\log t_i(s_i)$ are concave in s_i, MAP estimation reduces to a convex program. Another class of sparsity potentials are of the *Student's t* type:

$$t_i(s_i) = (1 + (\tau_i/\nu_i)\sigma^{-2}s_i^2)^{-(\nu_i+1)/2}, \quad \tau_i, \nu_i > 0. \tag{3}$$

For these, $\log t_i(s_i)$ is neither concave nor convex, and MAP estimation is in general not a convex program. In comparison with Laplace sites, Student's t potentials have heavier tails, and for small ν_i enforce sparsity more strongly. We refer to a model with likelihood (1) and non-Gaussian sparsity prior $P(\boldsymbol{u})$ as *sparse linear model* (SLM).

Beyond sparsity prior sites, our framework can be used for other models, whose posterior can be written in the form $P_0(\boldsymbol{u}) \prod_i t_i(s_i)$, where $P_0(\boldsymbol{u})$ has Gaussian form (but need not be normalizable), and the t_i are scalar potentials. For example, sparsity sites can be used as likelihood terms, to drive robust regression. Or the $t_i(s_i)$ may be binary classification likelihoods. Concrete examples will be given below.

2.1 Variational Lower Bounds

Bayesian inference is not analytically tractable in general for sparse linear models (SLMs) with non-Gaussian sparsity potentials $t_i(s_i)$, and has to be approximated. As motivated in Section 1, we are interested in scalable relaxations to standard linear optimization primitives, which is why we focus on a *variational* approach here, with roots in statistical physics.[2] Our restriction to SLMs with priors of factorizable form facilitates the exposition. Generalizations to non-Gaussian likelihoods and coupled non-Gaussian sites will be given in later sections.

The log partition function $\log P(\boldsymbol{y}) = \log \int P(\boldsymbol{y}|\boldsymbol{u})P(\boldsymbol{u})\, d\boldsymbol{u}$ contains the gist of the posterior,[3] and is easier to approximate. It cannot be computed exactly for the SLM, whose posterior is not Gaussian. At this point, we exploit a property of sparsity potentials mentioned above already: a positive even continuous function $t_i(s_i)$ is *(strongly) super-Gaussian* if $g_i(x_i) = \log t_i(s_i)$, $x_i = s_i^2/\sigma^2$, is convex and nonincreasing[4] for $x_i > 0$ (Palmer

[2] This does not mean that other approximation techniques, such as Markov chain Monte Carlo, cannot be scalable, only that equivalent relaxations to provably *few* calls of standard primitives are harder to establish.

[3] $\log P(\boldsymbol{y})$ is the cumulant generating function of $P(\boldsymbol{u}|\boldsymbol{y})$, with a role similar to the generating function for a convergent series, or the characteristic function for a distribution.

[4] Fenchel duality works for any convex $g_i(x_i)$, but if g_i is not nonincreasing, the maximization would not be over $\pi_i > 0$, so that at least for some x_i, the closest lower bound to $t_i(s_i)$ would not have Gaussian form (which requires $\pi_i > 0$). However, since $t_i(s_i) = e^{g_i(s_i^2/\sigma^2)}$ is in general a normalizable potential, we can constrain $g_i(x_i)$ to be nonincreasing without much loss of generality.

et al., 2006). We can represent this convex function using Fenchel[5] duality (Rockafellar, 1970, Sect. 12): $g_i(x_i) = \max_{\pi_i>0} -x_i\pi_i/2 - g_i^*(-\pi_i/2)$, resulting in

$$t_i(s_i) = \max_{\pi_i>0} e^{-\frac{1}{2}\sigma^{-2}\pi_i s_i^2} f_i(\pi_i), \quad f_i(\pi_i) = e^{-g_i^*(-\pi_i/2)}.$$

Note that $\log f_i(\pi_i)$ is concave, since the conjugate function g_i^* is convex just as g_i. The term "super-Gaussian" becomes clear now: $t_i(s_i)$ has tight Gaussian-form lower bounds of all possible widths.

Palmer et al. (2006) remark several interesting facts about this "Gaussianification" step. First, there is a close relationship to scale mixture decompositions (Gneiting, 1997; West, 1987), where a non-Gaussian density $t_i(s_i)$ is written as a mixture of zero mean Gaussians: $t_i(s_i) = \mathrm{E}_{\pi_i}[N(s_i|0, \pi_i^{-1}\sigma^2)]$. It is shown in (Palmer et al., 2006) that all scale mixture sites can be lower bounded as above, although $f_i(\pi_i)$ is different from the mixture density. Moreover, they show that the variational approximation that arises from these lower bounds, to be detailed right below, is equivalent for scale mixture sites to a different variational principle, known as *variational (mean field) Bayes*. More details about these relationships are found in (Palmer et al., 2006; Seeger, 2008), together with references to earlier work exploiting special cases of Fenchel duality lower bounds (Girolami, 2001; Figueiredo, 2003; Jaakkola, 1997; Wipf et al., 2004). The lower bounds for the Laplace (2) are

$$e^{-\tilde{\tau}_i|s_i|} = \max_{\pi_i>0} N^U(s_i|0, \sigma^{-2}\pi_i) e^{-(\tau_i^2/2)\pi_i^{-1}}, \quad \tilde{\tau}_i = \tau_i/\sigma, \tag{4}$$

where we define *unnormalized Gaussian* functions as

$$N^U(\boldsymbol{z}|\boldsymbol{b}, \boldsymbol{P}) := \exp\left(-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{P}\boldsymbol{z} + \boldsymbol{b}^T\boldsymbol{z}\right), \ \boldsymbol{P} \succeq \boldsymbol{0}.$$

For Student's t sites (3), we obtain

$$(1 + (\tau_i/\nu_i)\sigma^{-2}s_i^2)^{-(\nu_i+1)/2} = \max_{\pi_i>0} N^U(s_i|0, \sigma^{-2}\pi_i) e^{-h_i(\pi_i)/2}, \tag{5}$$

where $h_i(\pi_i) = (\nu_i + 1)(x_i - \log(1 + x_i))\mathrm{I}_{\{x_i \le 0\}}$ with $x_i = (\alpha_i/(\nu_i + 1))\pi_i - 1$, $\alpha_i = \nu_i/\tau_i$. Finally, if $t_i(s_i)$ is super-Gaussian, so is a positive mixture of the form

$$\sum_l \alpha_l t_i(\beta_l s_i), \quad \alpha_l, \beta_l > 0,$$

because the logsumexp function $\boldsymbol{x} \mapsto \log \boldsymbol{1}^T \exp(\boldsymbol{x})$ is strictly convex and increasing in each argument, and the log-mixture is the concatenation of logsumexp and $(\log t_i(\beta_l s_i) + \log \alpha_l)_l$, the latter being convex and nonincreasing for $x_i > 0$ in each component (Boyd and Vandenberghe, 2002, Sect. 3.2.4).

The variational relaxation we use here is obtained by plugging the Gaussian-form bounds into $P(\boldsymbol{y})$, which results in the lower bound

$$P(\boldsymbol{y}) \ge \tilde{C}_1 \int N(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{u}, \sigma^2\boldsymbol{I}) N^U(\boldsymbol{u}|\boldsymbol{0}, \sigma^{-2}\boldsymbol{B}^T\boldsymbol{\Pi}\boldsymbol{B})\, d\boldsymbol{u}, \quad \tilde{C}_1 = \prod_{i=1}^q f_i(\pi_i), \ \boldsymbol{\Pi} = \operatorname{diag}\boldsymbol{\pi}. \tag{6}$$

The right hand side is a Gaussian integral and can be evaluated easily. The variational problem, to be addressed by our algorithms, is to maximize the lower bound w.r.t. the variational parameters $\boldsymbol{\pi} \succ \boldsymbol{0}$ (*i.e.* , $\pi_i > 0$ for all $i = 1, \ldots, q$), with the aim of tightening the approximation to $\log P(\boldsymbol{y})$.

The idea of Gaussian-form lower bounding is not restricted to symmetric sites. For example, if $t_i(s_i) = \tilde{t}_i(s_i)e^{\alpha_i s_i}$, so that $\tilde{t}_i(s_i)$ is super-Gaussian, we can bound $\tilde{t}_i(s_i)$ as before, then multiply $e^{\alpha_i s_i}$ back (which is just log-linear). In the following, commiting a slight abuse of nomenclature, we will refer to such $t_i(s_i)$ as *super-Gaussian*. For example, Bernoulli potentials used in binary classification are

$$t_i(s_i) = \left(1 + e^{-y_i s_i}\right)^{-1} = \frac{e^{y_i s_i/2}}{2\cosh(y_i s_i/2)}, \tag{7}$$

[5] Under some additional conditions on $g_i(x_i)$, Fenchel duality is equivalent to Legendre duality.

and $-\log\cosh(y_i s_i/2)$ is even and convex as function of s_i^2 (Jaakkola, 1997, Sect. 3.B). The corresponding conjugate function is hard to compute analytically, but this is not required in our algorithms, as long as $g_i(x_i)$ and its derivative can be computed at any point (see Section 4.3).

In the following, we will address two major problems for this relaxation. In the next section, we will answer the question under which conditions the variational problem of maximizing the lower bound of (6) is a convex optimization problem. To this end, we will derive some novel convexity results for certain log determinants. Our findings have impact on MAP estimation algorithms as well (see Section 5.2).

However, we will see in Section 4 that the variational problem even in a convex case is more difficult to solve than the corresponding MAP estimation. For the latter, gradient computations come at the cost of solving a single linear system, while computing a gradient of the variational lower bound w.r.t. $\boldsymbol{\pi}$ is much more difficult than a single system. Most previous algorithms for variational problems of this kind (Girolami, 2001; Tipping, 2001; Minka, 2001) avoid this difficulty by following a step-wise approach, optimizing w.r.t. single components of $\boldsymbol{\pi}$ in turn, keeping all others fixed. While an informed scheduling of updates (Tipping and Faul, 2003; Seeger and Nickisch, 2008) can render these algorithms feasible on problems of moderate size, their scalability is fundamentally limited.[6] In difficult image reconstruction settings, every site approximation $N^U(s_i|0, \sigma^{-2}\pi_i)$ has to be visited at least once (usually several times). But for large n in the hundred thousands, every single scalar update requires the equivalent of a reweighted least squares estimation. And for problems considered here, where q can be a million or more, $O(q)$ LS estimations simply cannot be done.

Our second contribution lies in the development of novel classes of algorithms that can cope with such large scale problems, by decoupling the lower bound criterion complexity in a nested double loop fashion. These algorithms can be applied in domains where single-site updating is not an option. Also on problems of moderate size, speedups by orders of magnitude are realized. These algorithms can be applied in general, whether the variational problem is convex or not.

3 Convexity Properties of Variational Inference

We discussed a general variational relaxation of Bayesian inference for generalized sparse linear models in the previous section, to the maximization of the right hand side of (6). In this section, we bring this optimization problem into a form more amenable to algorithmic treatment. The relaxation is a special case of variational (mean field) Bayes (Ghahramani and Beal, 2001; Attias, 2000), or of direct site bounding (Jordan et al., 1997). We answer the question under which conditions the variational problem is a convex optimization problem. To this end, we prove a number of novel convexity properties for parts of the upper bound criterion, thereby laying groundwork for our novel scalable algorithms introduced in Section 4 as well.

Our problem is to maximize the right hand side of (6). Assume for now that $\boldsymbol{B}^T\boldsymbol{\Pi}\boldsymbol{B}$ is invertible. The end result remains valid even if this is not the case, as is easily seen by a continuity argument. Let $Q(\boldsymbol{u}) := C_2^{-1}N^U(\boldsymbol{u}|\boldsymbol{0}, \sigma^{-2}\boldsymbol{B}^T\boldsymbol{\Pi}\boldsymbol{B})$ and $Q(\boldsymbol{y}, \boldsymbol{u}) := P(\boldsymbol{y}|\boldsymbol{u})Q(\boldsymbol{u})$. The joint distribution is Gaussian, and

$$Q(\boldsymbol{u}|\boldsymbol{y}) = N(\boldsymbol{h}, \sigma^2\boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma}^{-1} = \boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\boldsymbol{\Pi}\boldsymbol{B}. \tag{8}$$

We have that

$$\int P(\boldsymbol{y}|\boldsymbol{u})Q(\boldsymbol{u})\,d\boldsymbol{u} = Q(\boldsymbol{y}) = |2\pi\sigma^2\boldsymbol{\Sigma}|^{1/2}\max_{\boldsymbol{u}} Q(\boldsymbol{y})Q(\boldsymbol{u}|\boldsymbol{y}) = |2\pi\sigma^2\boldsymbol{\Sigma}|^{1/2}\max_{\boldsymbol{u}} P(\boldsymbol{y}|\boldsymbol{u})Q(\boldsymbol{u}), \tag{9}$$

because the maximum of the Gaussian $Q(\boldsymbol{u}|\boldsymbol{y})$ is attained at the mean $\boldsymbol{h}$. The crucial step here is that we can move from the integral over $\boldsymbol{u}$ to the maximum over $\boldsymbol{u}$ *exactly*, which is possible because Q is Gaussian. We end up with $P(\boldsymbol{y}) \geq C_1 e^{-\phi(\boldsymbol{\pi})/2}$, where

$$\begin{aligned}
\phi(\boldsymbol{\pi}) &:= \log|\boldsymbol{A}| + h(\boldsymbol{\pi}) + \sigma^{-2}\left(\min_{\boldsymbol{u}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{u}\|^2 + \boldsymbol{s}^T\boldsymbol{\Pi}\boldsymbol{s}\right), \\
h(\boldsymbol{\pi}) &= \sum_{i=1}^q h_i(\pi_i) = -2\sum_{i=1}^q \log f_i(\pi_i),
\end{aligned} \tag{10}$$

[6] These comments apply to sparse Bayesian *inference*. For sparse *estimation*, forward selection schedules may well work for very large n and q, namely because many of the π_i are clamped to ∞ and never once moved from there (see Section 5.2).

and $C_1 = (2\pi\sigma^2)^{(n-m)/2}$. The variational problem now consists in maximizing the lower bound, or equivalently minimizing $\phi(\boldsymbol{\pi})$, w.r.t. the variational parameters $\boldsymbol{\pi}$. The Gaussian posterior approximation is $Q(\boldsymbol{u}|\boldsymbol{y})$, with the final parameters $\boldsymbol{\pi}$ plugged in. An algorithmically beneficial side effect of using a lower *bound* on $\log P(\boldsymbol{y})$, rather than just an approximation (as other variational methods, such as expectation propagation (Minka, 2001) do), is that we can usually devise algorithms which provably converge to a local optimum. Still, $\phi(\boldsymbol{\pi})$ is in general a coupled, non-convex function of non-standard form, and even $\nabla_{\boldsymbol{\pi}}\phi$ is hard to compute.

3.1 Some Convexity Results

In this section, we establish some convexity properties of ϕ in (10). A crucial term in this criterion is $\log|\boldsymbol{A}|$, where $\sigma^2\boldsymbol{A}^{-1}$ is the covariance matrix of the approximate posterior (8). At least in hindsight, much of the computational difficulty is caused by this term (see Section 5.2 for a detailed discussion). Let $\boldsymbol{\gamma} := \boldsymbol{\pi}^{-1}$,*i.e.* $\gamma_i = 1/\pi_i$. It turns out that $\log|\boldsymbol{A}|$ has a number of convexity properties in terms of $\boldsymbol{\pi}$ or $\boldsymbol{\gamma}$, which are important for obtaining scalable algorithms, or characterizations of the variational problem in the first place.

Theorem 1 *Let $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, $\boldsymbol{B} \in \mathbb{R}^{q\times n}$ be arbitrary matrices, so that*

$$\log|\boldsymbol{A}|, \quad \boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\boldsymbol{\Pi}\boldsymbol{B},\ \boldsymbol{\Pi} = \operatorname{diag}\boldsymbol{\pi}$$

exists for all $\boldsymbol{\pi} \succ \boldsymbol{0}$ from an open set. Let $\boldsymbol{\gamma} := \boldsymbol{\pi}^{-1}$, $\boldsymbol{\Gamma} = \operatorname{diag}\boldsymbol{\gamma}$.

1. *$\boldsymbol{\pi} \mapsto \log|\boldsymbol{A}|$ is concave.*
2. *$\boldsymbol{\gamma} \mapsto \log|\boldsymbol{A}|$ is convex.*
3. *$\boldsymbol{\gamma} \mapsto \log|\boldsymbol{A}| + \log|\boldsymbol{\Gamma}|$ is concave.*
4. *Let $\rho_i(\gamma_i)$ be concave functions into $\mathbb{R}_+$. Then, $\boldsymbol{\pi} \mapsto \log|\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\rho(\boldsymbol{\Pi})\boldsymbol{B}|$ is concave, where $\rho(\boldsymbol{\Pi}) = \operatorname{diag}(\rho_i(\pi_i))$.*
5. *Let $\rho_i(\gamma_i)$ be twice continuously differentiable functions into $\mathbb{R}_+$, so that*

   $$\rho_i''(\gamma_i)\rho_i(\gamma_i) \geq (\rho_i'(\gamma_i))^2$$

   for all i and γ_i. Then, $\boldsymbol{\gamma} \mapsto \log|\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\rho(\boldsymbol{\Gamma})\boldsymbol{B}|$ is convex, where $\rho(\boldsymbol{\Gamma}) = \operatorname{diag}(\rho_i(\gamma_i))$.
6. *Let $\rho_i(\gamma_i)$ be concave functions into $\mathbb{R}_+$. Then, $\boldsymbol{\gamma} \mapsto \log|\rho(\boldsymbol{\Gamma})| + \log|\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\rho(\boldsymbol{\Gamma})^{-1}\boldsymbol{B}|$ is concave, where $\rho(\boldsymbol{\Gamma}) = \operatorname{diag}(\rho_i(\gamma_i))$.*
7. *Let $Q(\boldsymbol{u}|\boldsymbol{y})$ be the approximate posterior of (8). Then,*

   $$\sigma^{-2}\mathrm{Var}_Q[s_i|\boldsymbol{y}] = \boldsymbol{\delta}_i^T\boldsymbol{B}\boldsymbol{A}^{-1}\boldsymbol{B}^T\boldsymbol{\delta}_i \leq \gamma_i.$$

For the proof, (1.) is well known (Boyd and Vandenberghe, 2002, Sect. 3.1.5). The generalization (4.) follows from (Boyd and Vandenberghe, 2002, Sect. 3.2.4), since $\boldsymbol{\pi} \mapsto \log|\boldsymbol{A}|$ is nondecreasing in each π_i. (2.) follows from the more general (5.), using $\rho_i(\gamma_i) = \gamma_i^{-1}$, and (3.) follows from (6.), using $\rho_i(\gamma_i) = \gamma_i$. To our knowledge, (5.) and (6.) are novel, at least we do not know of previous appearences in machine learning. They are proved in Appendix A.1. The proof of (7.) is a part of the proof of (5.).

Note that (5.) is more general than what we require in the following. For example, it holds for all $\rho_i(\gamma_i) = \gamma_i^{-\beta_i}$, $\beta_i > 0$. For $\rho_i(\gamma_i) = e^{\gamma_i}$, we obtain the convexity of $\boldsymbol{\gamma} \mapsto \log|\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\exp(\boldsymbol{\Gamma})\boldsymbol{B}|$, generalizing the *logsumexp* function $\boldsymbol{x} \mapsto \log \boldsymbol{1}^T\exp(\boldsymbol{x})$ (Boyd and Vandenberghe, 2002, Sect. 3.1.5) to matrix values. The convexity of the latter is behind many properties of exponential families or of maximum-likelihood estimation in log-linear models. Note also that (7.) gives a precise characterization of γ_i as sparsity parameter regulating the variance of s_i. A similar argument shows that the size of $\mathrm{E}_Q[s_i|\boldsymbol{y}]$ is also regulated by γ_i.What about the remaining terms in (10)? Here and in the following, we treat ϕ, h, and f_i as functions[7] of $\boldsymbol{\pi}$ or $\boldsymbol{\gamma}$. Based on Theorem 1, we show that $\boldsymbol{\gamma} \mapsto \phi(\boldsymbol{\gamma}) - h(\boldsymbol{\gamma})$ is a convex function.

[7] In general, we adopt the physics convention of treating function values as dependent variables, invariant to reparameterizations of the variables they depend on.

Theorem 2 *Our variational relaxation of approximate inference requires the minimization of $\phi(\boldsymbol{\gamma})$ from (10). The function $\boldsymbol{\gamma} \mapsto \phi(\boldsymbol{\gamma}) - h(\boldsymbol{\gamma})$, the negative log partition function of a Gaussian, is convex for $\boldsymbol{\gamma} \succ \mathbf{0}$.*

The convexity of $\log|\boldsymbol{A}|$ has been shown in Theorem 1, part (2.). $\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u}\|^2$ is convex in $\boldsymbol{u}$. More interestingly, $(\boldsymbol{u},\boldsymbol{\gamma}) \mapsto \boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}$ is *jointly* convex, since the *quadratic-over-linear* function $(s_i,\gamma_i) \mapsto s_i^2/\gamma_i$ is jointly convex for $\gamma_i > 0$ (Boyd and Vandenberghe, 2002, Sect. 3.1.5), and $\boldsymbol{s} = \boldsymbol{B}\boldsymbol{u}$ is linear in $\boldsymbol{u}$. Now, since $\min_{\boldsymbol{u}} \kappa(\boldsymbol{u},\boldsymbol{\gamma})$ is convex if κ is jointly convex (Boyd and Vandenberghe, 2002, Sect. 3.2.5), we conclude that $\boldsymbol{\gamma} \mapsto \phi(\boldsymbol{\gamma}) - h(\boldsymbol{\gamma})$ is a convex function.

To put this result into context, note that $\phi - h$ is the negative log partition function of a Gaussian with natural parameters being linear in $\boldsymbol{\pi}$. It is therefore well known that $\boldsymbol{\pi} \mapsto \phi - h$ is a *concave* function. However, $\boldsymbol{\pi} \mapsto h$ is convex ($h_i(\pi_i)$ is the conjugate function for $2g(x_i)$), which does not lead to insights about $\boldsymbol{\pi} \mapsto \phi$. The *convexity* of the negative log partition function of a distribution with natural parameters linear in $\boldsymbol{\gamma}^{-1}$ seems specific to the Gaussian case, and is novel to our knowledge.

Given Theorem 2, if all $h_i(\gamma_i)$ are convex, the whole variational problem $\min_{\boldsymbol{\gamma}\succ\mathbf{0}} \phi$ is a convex minimization. In the next section, we establish properties for when this is the case.

3.2 Convex Variational Inference

The statement of Theorem 2 implies that whenever $\boldsymbol{\gamma} \mapsto h(\boldsymbol{\gamma})$ is convex, the whole variational problem $\min_{\boldsymbol{\gamma}\succ\mathbf{0}} \phi$ of interest here is a convex problem with a unique solution. In this section, we show that whenever $\log t_i(s_i)$ are concave functions, and $t_i(s_i)$ are super-Gaussian (see Section 2.1), $h(\boldsymbol{\gamma})$ is in fact convex. Moreover, at least in general, this property is necessary for the convexity of $h(\boldsymbol{\gamma})$, and for the convexity of the variational problem. For notational simplicity, we do not deal with the most general case our result applies for here. For example, $P(\boldsymbol{y}|\boldsymbol{u})$ can be replaced by any factor of Gaussian form in $\boldsymbol{u}$, and the $t_i(s_i)$ can be likelihood sites, or depend on multiple components of $\boldsymbol{s}$. Generalizations are given in Section 5.1.

Theorem 3 *Consider a model with Gaussian likelihood (1) and a prior $P(\boldsymbol{u}) \propto \prod_{i=1}^q t_i(s_i)$, $\boldsymbol{s} = \boldsymbol{B}\boldsymbol{u}$. Let each $t_i(s_i)$ be strongly super-Gaussian, meaning that $t_i(s_i) = e^{\alpha_i s_i}\tilde{t}_i(s_i)$, where $\tilde{t}_i(s_i)$ is an even function, and $g_i(x_i) = \log\tilde{t}_i(s_i)$, $x_i = s_i^2$, is strictly[8] convex and nonincreasing for $x_i > 0$. Moreover, suppose that $g_i(s_i) = \log\tilde{t}_i(s_i)$ is concave[9] and twice continuously differentiable for $s_i > 0$. Then, the variational problem of minimizing (10) over $\boldsymbol{\gamma} \succ \mathbf{0}$ is a convex optimization problem.*

In general, the requirements on the $t_i(s_i)$ are necessary for the convexity to hold. If $g_i''(s_i) > 0$ for some $s_i > 0$, then $h_i(\gamma_i)$ is not convex at some $\gamma_i > 0$. For general $\boldsymbol{X}$, $\boldsymbol{B}$, and $\boldsymbol{y}$, this means that $\phi(\boldsymbol{\gamma})$ is not convex either.

The proof is given in Appendix A.2. Our theorem provides a satisfying characterization of the variational inference relaxation of Section 2. MAP estimation (see Section 1) is a convex problem if and only if (in general) all $\log t_i(s_i)$ are concave. Whenever MAP estimation is convex, and the $t_i(s_i)$ are super-Gaussian, the variational relaxation is convex as well. Loosely speaking, it is the log-concavity of the posterior that renders the variational problem convex. This property sets the relaxation used here apart from all other approximate inference methods for continuous variable models we know of: most of these can be shown to be non-convex in general, even if the log posterior is concave and has a single mode only.[10]

With a view to Section 4, it is also interesting to ask under which conditions the $h_i(\gamma_i)$ are concave functions. We do not have a complete characterization for this case, but can give some examples. If $t_i(s_i) = e^{-\tau|s_i|^\alpha}$, $\alpha \in (0,2]$, then $h_i(\gamma_i) \propto \gamma_i^\beta$ with $\beta = -\alpha/(\alpha-2)$, which is convex iff $\alpha \geq 1$. In these cases, $t_i(s_i)$ is log-concave. For $\alpha < 1$, $h_i(\gamma_i)$ is concave. Moreover, if $t_i(s_i)$ is *log-convex* in s_i, then

$$h_i(\gamma_i) = \max_{s_i} -\sigma^{-2}s_i^2/\gamma_i - 2\log t_i(s_i) = -\min_{s_i}\left(\sigma^{-2}s_i^2\gamma_i + 2\log t_i(s_i)\right),$$

which is concave in γ_i, since the argument of $\min_{s_i}$ is jointly convex in (s_i,γ_i) (Boyd and Vandenberghe, 2002, Sect. 3.2.5). Therefore, given that $t_i(s_i)$ is super-Gaussian, its log-{concavity/convexity} implies convexity/concavity of $h_i(\gamma_i)$. The reverse is not true in general (for the second statement), as can be seen for $t_i(s_i) = e^{-\tau|s_i|^\alpha}$ above.

[8] We require a slightly stronger notion of super-Gaussianity here, in that $g_i(x_i)$ has to be strictly convex.

[9] Here and elsewhere, we understand function values as variables dependent on their arguments, so that $g_i(s_i) = g_i(x_i)$. This convention, widely used in physics, simplifies notation and should not lead to confusions.

[10] An example is expectation propagation (Minka, 2001), whose log partition function approximation is non-convex (Opper and Winther, 2005), except for trivial cases. Log-concavity of the $t_i(s_i)$ has important consequences for the numerical properties of the EP algorithm (Seeger, 2008), but they do not imply convexity of the complete problem.

We close this section with some examples. For Laplace sites (2), the Fenchel duality is given by (4), where $h_i(\gamma_i) = \tau_i^2\gamma_i$, a convex function as predicted by our result above. For sparse linear models with Laplace sites, MAP estimation is a convex quadratic program, with the Lasso as a special case (see Section 2). Variational inference is a convex problem as well. While the same relaxation has been used before for these SLMs (Girolami, 2001), its convexity has not been established until now.

Second, binary classification Bernoulli likelihood sites (7), also known as logistic potentials, are super-Gaussian (see Section 2.1), and they are well known to be log-concave. MAP estimation for generalized linear models with these sites is known as logistic regression, a convex problem typically solved by the iteratively reweighted least squares (IRLS) algorithm (also known as Fisher scoring). Variational inference for this model is a convex problem, and our algorithms introduced in Section 4 make use of IRLS as well.

However, Student's t potentials (3) are not log-concave, and $h_i(\gamma_i)$ in (5) is neither convex nor concave. Neither MAP estimation nor variational approximate inference is a convex problem, when Student's t sites are used.

We have provided a satisfying characterization of a widely used class of variational approximate inference methods. For super-Gaussian sites, the variational problem is convex if and only if the search for the posterior mode is convex. This does not mean that solving the variational problem is computationally as tractable as MAP estimation (see end of Section 2.1). For example, our posthoc result that the algorithm of (Girolami, 2001) solves a convex problem, is of little value for measurement design on full-size images, where this algorithm would not converge in any reasonable amount of time. In the next section, we propose classes of algorithms that solve the variational problem in a scalable way. While they are still in general more expensive than convex MAP estimation methods, the precise relationship is clarified in Section 5.2.

4 The Algorithms

In this section, we develop two classes of algorithms for scalable variational inference, maximizing the lower bound of (6) for SLMs, where the prior is $P(\boldsymbol{u}) \propto \prod_{i=1}^{q} t_i(s_i)$, $\boldsymbol{s} = \boldsymbol{B}\boldsymbol{u}$, with super-Gaussian sites. We use Fenchel concave duality in order to upper bound the variational criterion by a decoupling surrogate, which can be minimized by a standard convex optimization algorithm.

These algorithms are independent of our convexity analysis for the variational problem in Section 3. They can be applied in order to find a local minimum very efficiently in general. If the problem is convex, there is a unique global optimum, which is found by our methods.

4.1 The First Class

We propose two closely related classes of scalable algorithms for minimizing ϕ of (10) w.r.t. $\boldsymbol{\pi} \succ \mathbf{0}$ (or equivalently, w.r.t. $\boldsymbol{\gamma} := \boldsymbol{\pi}^{-1} \succ \mathbf{0}$). In this section, we introduce the first class. Whether ϕ is convex in $\boldsymbol{\gamma}$ or not (see Theorem 3), it is not obvious how to minimize ϕ tractably, since even the computation of $\nabla_{\boldsymbol{\gamma}}\phi$ will be seen to be a computationally expensive problem.

We make use of a powerful general idea known as *double loop* algorithms, *concave-convex* algorithms, or *d.c. programming* (difference of convex). Special cases of such algorithms are already heavily used in machine learning and statistics: the expectation-maximization method (Dempster et al., 1977), variational (mean field) Bayesian inference (Attias, 2000), or CCCP for approximate inference (Yuille and Rangarajan, 2003), among many others. The idea is to write ϕ as sum of a concave and a convex part. We use Fenchel-Legendre duality once more, in order to upper bound the concave part by a linear function, then minimize the convex upper bound to ϕ globally. The linear upper bounding is done iteratively, in so called outer loop steps, followed by inner loop convex minimizations. If the concave part is differentiable, the linear upper bound is a tangent plane, and under mild conditions the resulting double loop algorithm can be shown to be globally convergent, meaning that it converges to a local minimum point of ϕ, no matter from where we start.

In Theorem 1, part (3.), we show that $\boldsymbol{\gamma} \mapsto \log|\boldsymbol{A}| + \log|\boldsymbol{\Gamma}|$ is a concave function. Now write $h(\boldsymbol{\gamma}) - \log|\boldsymbol{\Gamma}| = h_\cap(\boldsymbol{\gamma}) + h_\cup(\boldsymbol{\gamma})$, where $h_\cap$ is concave, and $h_\cup$ is convex. Note that $h(\boldsymbol{\gamma}) - \log|\boldsymbol{\Gamma}|$ decomposes as sum of scalar terms, each depending on a single γ_i only, and it should in general be possible to find $h_\cap$, $h_\cup$ which decompose in the same way. A concave-convex decomposition is never unique: we can always add a concave part to $h_\cap$ and subtract it from $h_\cup$. However, since we will replace the concave part by a hyperplane, it is sensible to choose a decomposition with "minimal" $h_\cap$, as close to linear as possible. For tractability, it is also important that both parts are simple terms composed of standard functions. For example, for Student's t sites (3), $h_i(\gamma_i)$ is given in (5), where $\boldsymbol{\alpha} = \boldsymbol{\nu} \circ \boldsymbol{\tau}^{-1}$. A convenient decomposition is $h_\cap(\boldsymbol{\gamma}) = \boldsymbol{\nu}^T(\log\boldsymbol{\gamma})$ and $h_{\cup,i}(\gamma_i) = \alpha_i/\gamma_i + (\nu_i +$

$1)(\log(\nu_i+1)-\log\alpha_i-1)$ for $\gamma_i \geq \alpha_i/(\nu_i+1)$, $h_{\cup,i}(\gamma_i) = -(\nu_i+1)\log\gamma_i$ for $\gamma_i < \alpha_i/(\nu_i+1)$. Note that the latter is convex and twice differentiable.

Now, if $g(\boldsymbol{\gamma}) := \log|\boldsymbol{A}| + \log|\boldsymbol{\Gamma}| + h_\cap(\boldsymbol{\gamma})$, which is concave as sum of concave functions, then

$$\phi(\boldsymbol{\gamma}) = g(\boldsymbol{\gamma}) + \min_{\boldsymbol{u}} h_\cup(\boldsymbol{\gamma}) + \sigma^{-2}\left(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u}\|^2 + \boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}\right).$$

Using the conjugate representation $g(\boldsymbol{\gamma}) = \min_{\boldsymbol{z}\succeq\boldsymbol{0}} \boldsymbol{z}^T\boldsymbol{\gamma} - g^*(\boldsymbol{z})$, we obtain

$$\phi(\boldsymbol{\gamma}) \leq \min_{\boldsymbol{u}} \phi_{\boldsymbol{z}}(\boldsymbol{u},\boldsymbol{\gamma}), \quad \phi_{\boldsymbol{z}}(\boldsymbol{u},\boldsymbol{\gamma}) := \boldsymbol{z}^T\boldsymbol{\gamma} + h_\cup(\boldsymbol{\gamma}) + \sigma^{-2}\left(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u}\|^2 + \boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}\right) - g^*(\boldsymbol{z})$$

We saw in the proof of Theorem 2 that $\phi_{\boldsymbol{z}}$ is jointly convex. Our algorithm to minimize ϕ iterates between outer loop updates of $\boldsymbol{z} \leftarrow \operatorname{argmin} \boldsymbol{z}^T\boldsymbol{\gamma} - g^*(\boldsymbol{z})$ and inner loop convex minimizations of $\phi_{\boldsymbol{z}}$, in order to update $\boldsymbol{\gamma}$. As a by-product, we obtain minimum points $\boldsymbol{u}$, and it is easy to see that at the end of an outer loop, $\boldsymbol{u}$ is the mean[11] of the current posterior approximation $Q(\cdot|\boldsymbol{y})$.

The inner loop minimization of $\phi_{\boldsymbol{z}}$ can be done in any order of $\boldsymbol{u}$ and $\boldsymbol{\gamma}$. If $h_\cup(\boldsymbol{\gamma}) = \sum_i h_{\cup,i}(\gamma_i)$, as is the case in general, it is easiest to perform the minimization over $\boldsymbol{\gamma}$ first, since this decouples into q independent minimizations

$$\min_{\gamma_i} z_i\gamma_i + h_{\cup,i}(\gamma_i) + \sigma^{-2}s_i^2/\gamma_i,$$

leading to the equations $z_i + (dh_{\cup,i})/(d\gamma_i) - \sigma^{-2}s_i^2\gamma_i^{-2} = 0$. For the examples given here, these equations can be solved analytically, but in general univariate convex minimization can be used in order to solve for the required quantities (see Section 4.3). Plugging the solutions in, we obtain convex functions $h^*_{\cup,i}(s_i)$. For the case of Student's t sites, we have $h_{\cup,i}(\gamma_i) = \alpha_i/\gamma_i$, therefore $h^*_{\cup,i}(s_i) = 2(z_ip_i)^{1/2}$, $p_i = \alpha_i + s_i^2/\sigma^2$ (recall that $\alpha_i = \nu_i/\tau_i$). The remaining inner loop problem is

$$\min_{\boldsymbol{u}} \sigma^{-2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u}\|^2 + \sum_{i=1}^{q} h^*_{\cup,i}(s_i), \quad \boldsymbol{s} = \boldsymbol{B}\boldsymbol{u}. \tag{11}$$

This problem is of standard form, and can be solved generically by the *iteratively reweighted least squares* (IRLS) algorithm, a variant of the Newton-Raphson method. This method, which is discussed in detail in Section 4.3, proceeds in Newton steps, each of which requires the solution of a single linear system with a matrix of the same form as $\boldsymbol{A}$ (8), only that $\boldsymbol{\Pi}$ is replaced by a different positive diagonal matrix, and a simple line search. Convergence is typically attained after few iterations (less than thirty), and each step is reduced to a single (reweighted) least squares problem. Therefore, the inner loop minimization fulfils our criteria of scalability.

The outer loop updates of $\boldsymbol{z}$, given $\boldsymbol{\gamma}$, require the minimization of $\boldsymbol{z}^T\boldsymbol{\gamma} - g^*(\boldsymbol{z})$ for the concave function $g^*(\boldsymbol{z})$. It is not in general possible to analytically obtain $g^*(\boldsymbol{z})$. By duality, $g^*(\boldsymbol{z})$ can at any point be evaluated by minimizing $\boldsymbol{z}^T\boldsymbol{\gamma} - g(\boldsymbol{\gamma})$ over $\boldsymbol{\gamma}$, which is a convex problem, but is hard in general. Fortunately, none of this is necessary. For fixed $\boldsymbol{\gamma}$, the minimizer $\boldsymbol{z}_{opt}$ is such that

$$\boldsymbol{z}_{opt}^T\boldsymbol{\gamma} - g(\boldsymbol{\gamma}) = g^*(\boldsymbol{z}_{opt}) = \min_{\tilde{\boldsymbol{\gamma}}} \boldsymbol{z}_{opt}^T\tilde{\boldsymbol{\gamma}} - g(\tilde{\boldsymbol{\gamma}}),$$

so that $\nabla_{\boldsymbol{\gamma}}\boldsymbol{z}_{opt}^T\boldsymbol{\gamma} - g(\boldsymbol{\gamma}) = \boldsymbol{z}_{opt} - \nabla_{\boldsymbol{\gamma}}g(\boldsymbol{\gamma}) = \boldsymbol{0}$. We just have to compute the gradient of g at $\boldsymbol{\gamma}$. Using the fact that $d\log|\boldsymbol{A}| = \operatorname{tr}\boldsymbol{\Sigma}(d\boldsymbol{A}) = \operatorname{tr}\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T(d\boldsymbol{\Pi})$, we arrive at

$$\boldsymbol{z}_{opt} = \nabla_{\boldsymbol{\gamma}}g(\boldsymbol{\gamma}) = \boldsymbol{\pi} + \nabla_{\boldsymbol{\gamma}}h_\cap(\boldsymbol{\gamma}) - \boldsymbol{\pi}^2 \circ \operatorname{diag}^{-1}\left(\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T\right).$$

Since $h_\cap(\boldsymbol{\gamma}) = \sum_i h_{\cap,i}(\gamma_i)$, the main difficulty is the computation of the diagonal of $\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T$. A glance at (8) reveals that we need to compute *Gaussian marginal variances*: $(\operatorname{Var}_Q[s_i\,|\,\boldsymbol{y}])_i = \sigma^2\operatorname{diag}^{-1}(\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T)$. This is the second primitive we require in our algorithms. As opposed to least squares estimation, this computation is not usually done in non-Bayesian applications. It is important to note that we require *all* marginal variances in parallel. A single variance could be obtained as the solution of a least squares problem, but computing $\boldsymbol{z}_{opt}$ by solving q such problems is certainly not feasible. Nevertheless, there are algorithms from numerical mathematics which can be used to estimate these variances, and they reduce to the same primitives as does least squares estimation. In

[11] We abuse notation slightly by treating $\boldsymbol{u}$ as variable to optimize over here.

practice, the number of outer loop updates required to gain convergence is small, usually five or six iterations are sufficient. The variance estimation problem is discussed in more detail in Section 4.5.

This concludes the specification of our first class of algorithms. To understand the considerable computational benefits of the double loop structure, note that a *single* gradient computation $\nabla_{\boldsymbol{\gamma}}\phi$ is as expensive as an outer loop update in our scheme: both require all q Gaussian marginal variances. By bounding the $\log|\boldsymbol{A}|$ part, which causes the complexity (see Section 5.2), we reduce the number of expensive steps required until convergence drastically, compared to standard gradient descent algorithms. We follow the double loop strategy even if $h(\boldsymbol{\gamma})$ is convex, since the decoupled inner loop criterion is much more efficient to minimize than ϕ itself. Some general characteristics of our algorithms are given in Section 4.4.

4.2 The Second Class

Our second class of algorithms is closely related to the first one, but involves another twist. Again, our aim is to devise a double loop scheme. From Theorem 1, part (1.), we know that $\boldsymbol{\pi} \mapsto \log|\boldsymbol{A}|$ is concave. We can try to decompose $h = h_\cap(\boldsymbol{\pi}) + h_\cup(\boldsymbol{\gamma})$ (recall that $\boldsymbol{\gamma} = \boldsymbol{\pi}^{-1}$), $h_\cap$ concave in $\boldsymbol{\pi}$, and $h_\cup$ convex in $\boldsymbol{\gamma}$. For example, for Laplace sites (4), we have that $h = (\boldsymbol{\tau}^2)^T\boldsymbol{\gamma}$, which is convex, so that $h_\cup(\boldsymbol{\gamma}) = (\boldsymbol{\tau}^2)^T\boldsymbol{\gamma}$ and $h_\cap(\boldsymbol{\pi}) = 0$.

We can now proceed much as for the first class above, but treating $g(\boldsymbol{\pi}) := \log|\boldsymbol{A}| + h_\cap(\boldsymbol{\pi})$ as concave function in $\boldsymbol{\pi}$. By Fenchel's inequality (Rockafellar, 1970, Sect. 12): $g(\boldsymbol{\pi}) \le \boldsymbol{z}^T\boldsymbol{\pi} - g^*(\boldsymbol{z})$ for $\boldsymbol{z} \succ \boldsymbol{0}$. But this upper bound is also a convex function of $\boldsymbol{\gamma} \succ \boldsymbol{0}$ (since $z_i > 0$ for all i), which is the additional observation we make use of here. We obtain the convex inner loop problem

$$\min_{\boldsymbol{\gamma}} \min_{\boldsymbol{u}} \boldsymbol{z}^T(\boldsymbol{\gamma}^{-1}) + h_\cup(\boldsymbol{\gamma}) + \sigma^{-2}\left(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{u}\|^2 + \boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}\right) - g^*(\boldsymbol{z}),$$

which is treated much as in Section 4.1. The independent minimizations are now $\min_{\gamma_i} h_{\cup,i}(\gamma_i) + (z_i + \sigma^{-2}s_i^2)/\gamma_i$. If $p_i = z_i + \sigma^{-2}s_i^2$, the stationary equations are $(dh_{\cup,i})/(d\gamma_i) - p_i\gamma_i^{-2} = 0$. Plugging the solutions in gives rise to convex functions $h^*_{\cup,i}(s_i)$, and the inner loop optimization is done by the IRLS algorithm. For Laplace sites, $h_{\cup,i}(\gamma_i) = \tau_i^2\gamma_i$, so that $h^*_{\cup,i}(s_i) = 2\tau_i p_i^{1/2}$. For the outer loop update,

$$\boldsymbol{z}_{opt} = \mathrm{diag}^{-1}\left(\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T\right) + \nabla_{\boldsymbol{\pi}} h_\cap(\boldsymbol{\pi}).$$

If $h_\cap$ decomposes, the main effort here is again to compute the Gaussian variances of $Q(\boldsymbol{s}|\boldsymbol{y})$.

Characteristics of the second class, as well as a comparison with the first, are given in Section 4.4. The main idea is the same as for the first class: difficult coupling terms in ϕ are bounded by simple decoupling functions, so that the complexity of computing $\nabla_{\boldsymbol{\gamma}}\phi$ is shifted into a few outer loop updates. If $R = \sigma^{-2}\min_{\boldsymbol{u}}[\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{u}\|^2 + \boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}]$, the relationship between the two classes is given by

$$\phi^{[1]}_{\boldsymbol{z}} \ge \underbrace{\log|\boldsymbol{A}| + \log|\boldsymbol{\Gamma}| + (h - \log|\boldsymbol{\Gamma}|)}_{\boldsymbol{z}^T\boldsymbol{\gamma} - g^*(\boldsymbol{z}) + h^{[1]}_\cup \ge} + R = \phi = \underbrace{\log|\boldsymbol{A}| + h}_{\le \boldsymbol{z}^T(\boldsymbol{\gamma}^{-1}) - g^*(\boldsymbol{z}) + h^{[2]}_\cup} + R \le \phi^{[2]}_{\boldsymbol{z}}.$$

The superscripts distinguishing the two classes will be dropped in the sequel, where it will always be clear from the context which class is used.

Hybrid Variant

The coupling term $\log|\boldsymbol{A}|$ is bounded in either of our classes, making use of Fenchel's inequality for concave functions. In cases where the second class does not apply, the following hybrid variant can be used. Suppose that $h(\boldsymbol{\gamma}) = h_\cap(\boldsymbol{\gamma}) + h_\cup(\boldsymbol{\gamma})$, where $h_\cap$ is concave, $h_\cup$ is convex. We can upper bound $\log|\boldsymbol{A}| \le \boldsymbol{z}^T(\boldsymbol{\gamma}^{-1}) - g^*(\boldsymbol{z})$ and $h_\cap(\boldsymbol{\gamma}) \le \boldsymbol{z}_\cap^T\boldsymbol{\gamma} - g^*_\cap(\boldsymbol{z}_\cap)$. Since $h_\cap(\boldsymbol{\gamma})$ decouples, $\boldsymbol{z}_\cap$ and $g^*_\cap(\boldsymbol{z}_\cap)$ are easy to compute. In the inner loop, γ_i is eliminated by $\min_{\gamma_i} h_{\cup,i}(\gamma_i) + p_i/\gamma_i + z_{\cap,i}\gamma_i$, where $p_i = z_i + \sigma^{-2}s_i^2$. For Student's t sites (see Section 4.1), we have that $h^*_{\cup,i}(s_i) = 2(z_{\cap,i}(\alpha_i + p_i))^{1/2}$, and finally $\gamma_i' = ((\alpha_i + p_i')/z_{\cap,i})^{1/2}$. Moreover, the outer loop update is $\boldsymbol{z}_\cap = (\boldsymbol{\nu} + \boldsymbol{1}) \circ \boldsymbol{\pi}$.

4.3 The Inner Loop Optimization

In this section, we show how to efficiently solve the inner loop problem (11). The structure of $\phi_{\boldsymbol{z}}(\boldsymbol{u})$ is the sum of a least squares term and a decomposing penalty. Therefore, the Newton-Raphson algorithm reduces to a standard

method called iteratively reweighted least squares (IRLS). We describe a single Newton step here, starting from $\boldsymbol{u}$. Let $\boldsymbol{r} := \boldsymbol{y} - \boldsymbol{X}\boldsymbol{u}$ denote the residual vector. Then,

$$d\sigma^{-2}\|\boldsymbol{r}\|^2 = -2\sigma^{-2}\boldsymbol{r}^T\boldsymbol{X}(d\boldsymbol{u}), \quad d^2\sigma^{-2}\|\boldsymbol{r}\|^2 = 2\sigma^{-2}(d\boldsymbol{u})^T\boldsymbol{X}^T\boldsymbol{X}(d\boldsymbol{u}).$$

If $\theta_i := (\sigma^2/2)(dh^*_{\cup,i})/(ds_i)$, $\rho_i := (\sigma^2/2)(d^2h^*_{\cup,i})/(ds_i^2)$, then

$$\begin{aligned}\boldsymbol{g} &:= \nabla_{\boldsymbol{u}}\phi_{\boldsymbol{z}} = -2\sigma^{-2}\left(\boldsymbol{X}^T\boldsymbol{r} - \boldsymbol{B}^T\boldsymbol{\theta}\right),\\ \boldsymbol{H} &:= \nabla\nabla_{\boldsymbol{u}}\phi_{\boldsymbol{z}} = 2\sigma^{-2}\left(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T(\operatorname{diag}\boldsymbol{\rho})\boldsymbol{B}\right).\end{aligned}$$

Note that $\rho_i \geq 0$, by the convexity of $h^*_{\cup,i}$. The Newton search direction is

$$\boldsymbol{d} := -\boldsymbol{H}^{-1}\boldsymbol{g} = \left(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T(\operatorname{diag}\boldsymbol{\rho})\boldsymbol{B}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{r} - \boldsymbol{B}^T\boldsymbol{\theta}\right).$$

The computation of $\boldsymbol{d}$ requires to solve a system with the matrix $\boldsymbol{H}$, which is of the same form as $\boldsymbol{A}$. This is precisely the computation required for least squares estimation with the likelihood $P(\boldsymbol{y}|\boldsymbol{u})$ and the Gaussian prior $N(\boldsymbol{s}|\boldsymbol{0}, (\operatorname{diag}\boldsymbol{\rho})^{-1})$. Such systems are generally solved approximately by the linear conjugate gradients (LCG) algorithm (Golub and Van Loan, 1996). The cost per iteration of LCG is dominated by a MVM with $\boldsymbol{H}$, which translates to single MVMs with $\boldsymbol{X}$, $\boldsymbol{B}$, and its transposes respectively. Our scalability requirements are therefore met.

A line search along $\boldsymbol{d}$ can be run in negligible time. If $f(t) := (\sigma^2/2)\phi_{\boldsymbol{z}}(\boldsymbol{u} + t\boldsymbol{d})$, then

$$f'(t) = -(\boldsymbol{X}\boldsymbol{d})^T\boldsymbol{r} + t\|\boldsymbol{X}\boldsymbol{d}\|^2 + (\boldsymbol{B}\boldsymbol{d})^T\boldsymbol{\theta}^{(t)}.$$

Here, $\boldsymbol{\theta}^{(t)}$ is in terms of $\boldsymbol{s}^{(t)} = \boldsymbol{s} + t\boldsymbol{B}\boldsymbol{d}$. If we precompute $\boldsymbol{X}\boldsymbol{d}$, $(\boldsymbol{X}\boldsymbol{d})^T\boldsymbol{r}$, $\|\boldsymbol{X}\boldsymbol{d}\|^2$, and $\boldsymbol{B}\boldsymbol{d}$, $f(t)$ and $f'(t)$ can be evaluated in $O(q)$. No further MVMs are required during the line search. Each line search is started with $t_0 = 1$. Note that a line search seems essential in practice. Especially in the beginning, or after $\boldsymbol{X}$ has just been extended in a sequential design loop (see Section 6), a full Newton step ($t = 1$) would lead to a large *increase* of the criterion, and significantly smaller t need to be taken. During later stages, the first choice $t = 1$ is usually accepted, due to the self-scaling properties of the Newton method. Finally, once $\boldsymbol{u}' = \operatorname{argmin}\phi_{\boldsymbol{z}}(\boldsymbol{u})$ is found, $\boldsymbol{\gamma}'$ is updated as minimizer w.r.t. $\boldsymbol{\gamma}$, solving the scalar stationary equations once more.

For the case of Laplace sites and the second class, recall that $h^*_{\cup,i}(s_i) = 2\tau_i p_i^{1/2}$, $p_i = z_i + \sigma^{-2}s_i^2$. Therefore, $\theta_i = \tau_i s_i p_i^{-1/2}$ and $\rho_i = \tau_i z_i p_i^{-3/2}$. For Student's t sites, $h^*_{\cup,i}$ has the same form, but τ_i^2 is replaced by z_i, and z_i by $\alpha_i = \nu_i/\tau_i$. Finally, if $h^*_{\cup,i}(s_i)$ cannot be determined analytically, the procedure detailed in Appendix A.3 can be used, which does not increase the computational complexity. In some cases, even $h_i(\gamma_i)$ may not be known explicitly, or may be cumbersome to obtain analytically (see Section 2.1). We show in Appendix A.3 how our algorithms can be run based on computations of $g_i(x_i)$ and its derivative only.

4.4 Properties of the Algorithms

In this section, we analyze characteristics of the two classes of algorithms, relating them to each other, and showing how they compare with the sparse estimation method of (Wipf and Nagarajan, 2008), which inspired some of our work here. Properties of the variational problem they adress, are analyzed in Section 3.

First, the algorithms in either class can be shown to converge globally, *i.e.* to find a local minimum point from any starting point, if the outer loop updates are done exactly. This is seen just as in (Wipf and Nagarajan, 2008), whose arguments apply here as well. If the variational problem itself is convex (see Section 3), the algorithms converge to the global optimum from any starting point. In a nutshell, the argument goes as follows. The upper bound $\phi_{\boldsymbol{z}}$ has the same tangent plane at $\boldsymbol{\gamma}$ than ϕ. Therefore, the inner loop optimization is guaranteed to decrease ϕ substantially if $\nabla_{\boldsymbol{\gamma}}\phi$ is not equal to zero. Note that the convergence proof does not require the inner loop optimization to find the minimum of $\phi_{\boldsymbol{z}}$. In fact, a single line search along the first Newton direction would be sufficient. On the other hand, the outer loop updates have to be done to high accuracy to retain the guarantee, which can be problematic in large scale settings. This point is discussed in more detail in Section 4.5.

Our first class of algorithms can be seen as generalization of the sparse estimation method in (Wipf and Nagarajan, 2008). While they considered the special case $\boldsymbol{B} = \boldsymbol{I}$ only, their method can be generalized to any $\boldsymbol{B}$, using the facts proved in Theorem 1. It is obtained as limit of the Student's t case in Section 4.1, setting $\nu_i = 0$ for all

i. In this case, referred to as *automatic relevance determination* (ARD), the prior sites $t_i(s_i)$ are not normalizable. Moreover, as can be seen from Section 4.3, their inner loop criterion becomes

$$\sigma^{-2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{u}\|^2 + 2\sigma^{-1}\sum_{i=1}^{q} z_i^{1/2}|s_i|.$$

Solving for $\boldsymbol{u}$ is a quadratic program, whose special case for $\boldsymbol{B} = \boldsymbol{I}$ is the Lasso (see Section 2). Due to the nondifferentiable criterion, special code is required (but see Section 5.2), while our inner loops can be implemented more easily. On the other hand, for the inner loop solution $\boldsymbol{u}'$, many components in $\boldsymbol{s}' = \boldsymbol{B}\boldsymbol{u}'$ are exactly zero, and this fact can be exploited to find $\boldsymbol{u}'$ very efficiently. Moreover, much effort is concentrated on designing fast algorithms for Lasso, at least for certain $\boldsymbol{B}$ (unfortunately the most studied case, $\boldsymbol{B} = \boldsymbol{I}$, for which very efficient soft thresholding algorithms are known, is not very useful in the context of natural images). The exact sparsity of $\boldsymbol{s}'$ is even more useful for computing outer loop updates efficiently. Namely, since $\gamma_i' = \sigma^{-1} z_i^{-1/2}|s_i'|$, we see that $\boldsymbol{\gamma}$ is exactly sparse just as $\boldsymbol{s}'$. If there are d non-zeros in $\boldsymbol{\gamma}$, the computation[12] of $\boldsymbol{z}_{opt}$ can be done based on a system matrix of size $d \times d$. However, this computationally beneficial role of sparsity can be expected in the sparse estimation context only. This important point is discussed in more detail in Section 5.2. Variational approximate inference, with the aim of a useful uncertainty representation, is harder to do in practice than sparse estimation, and we show how the added complexity can be addressed computationally.

How generally applicable is either class of algorithms? First, we require a Gaussian-form lower bound on $t_i(s_i)$. If the site is super-Gaussian, Fenchel duality provides a tight lower bound. Non-symmetric sites can also be dealt with in our framework, as long as lower bounds are known (see Section 2.1 for binary classification Bernoulli sites). If they make use of two parameters,*i.e.* are of the form $t_i(s_i) \geq N^U(s_i|b_i, \pi_i) f_i(\pi_i, b_i)$, the extension given in Appendix A.4 may be applicable.

Second, in order to apply the first class, we require a decomposition of $h(\boldsymbol{\gamma}) - \log|\boldsymbol{\Gamma}|$ into a sum of concave and convex functions. Such a decomposition can always be found in principle, although we also require that the parts have a simple tractable form, and are at least differentiable. In all cases of convex $\log t_i(\sqrt{\cdot})$ we know of, this second requirement does not limit the applicability. We have seen above that $h^*_{\cup,i}(s_i)$ may not be differentiable, which precludes the direct usage of IRLS for solving the inner loop problems. However, ARD above is the only case we know of where this happens, and the ARD sites $t_i(s_i)$ do not correspond to normalizable priors.

On the other hand, in order to apply the second class of algorithms, $h(\boldsymbol{\gamma})$ has to be decomposed as $h_\cap(\boldsymbol{\pi}) + h_\cup(\boldsymbol{\gamma})$. This is simple if $h(\boldsymbol{\gamma})$ is convex itself, since $h_\cap(\boldsymbol{\pi}) = 0$ then. Laplace sites (2) and Bernoulli potentials (7), as well as all super-Gaussian, log-concave potentials, can be treated this way. However, a simple decomposition does not seem to exist in many other cases.

Note that whenever $h(\boldsymbol{\gamma})$ is convex, the first class of algorithms can be applied as well, using $h_\cup(\boldsymbol{\gamma}) = h(\boldsymbol{\gamma}) - \log|\boldsymbol{\Gamma}|$. However, in this case, the second class seems to be the more direct approach. For Laplace potentials, the inner loop criterion for the second class is more sparsity-enforcing than for the first.

The SLM can be configured with different sparsity prior sites. In this paper, Laplace and Student's t sites are treated explicitly. For the former, variational inference is convex. For the latter, the posterior is multi-modal, and our algorithms search for a local optimum. The choice of sparsity potentials should depend on the problem addressed.

How are the algorithms initialized? In practice, we found it useful to start with $\boldsymbol{z} = \varepsilon\boldsymbol{1}$ for some $\varepsilon > 0$ (say: $1/20$), and with $\boldsymbol{u} = \boldsymbol{0}$. We also explored the alternative of setting $\boldsymbol{\pi} \propto \boldsymbol{1}$, $\boldsymbol{u} = \boldsymbol{0}$, and start with an update of $\boldsymbol{z}$, but this led to less stable behaviour and required more running time.

4.5 Estimation of Gaussian Variances

Recall from Section 4.1 that the outer loop steps in our algorithms require the estimation of Gaussian variances $\tilde{\boldsymbol{z}} = \mathrm{diag}^{-1}(\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T) = \sigma^{-2}(\mathrm{Var}_Q[s_i|\boldsymbol{y}])_i$, which would also be required for the gradient computation $\nabla_{\boldsymbol{\gamma}}\phi$. Variance computations are not usually required in the context of sparse MAP estimation (see Section 5.2). Certainly, the vector $\tilde{\boldsymbol{z}}$ cannot be estimated by solving a single or few linear systems. In this section, we discuss a generic way of estimating these variances en bulk, using the Lanczos algorithm from numerical mathematics. Further details

[12] The computation of $\boldsymbol{z}_{opt}$ is slightly complicated if some $\gamma_i = 0$. From Section 4.1, we have that $z_i = \pi_i(1 - \pi_i\tilde{z}_i)$, where $\tilde{z}_i = 0$, and $\pi_i = \infty$. If the subscripts "0" ("1") denote the part of $\boldsymbol{\gamma}$ which is zero (non-zero), and $\boldsymbol{A}_1 = \boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}_1^T\boldsymbol{\Gamma}_1^{-1}\boldsymbol{B}_1$, a careful computation shows that $\boldsymbol{z}_0 = \mathrm{diag}^{-1}([\boldsymbol{B}_0\boldsymbol{A}_1^{-1}\boldsymbol{B}_0^T]^{-1})$. The z_i do not in general become zero.

are given in Appendix B. This section is technically more difficult than the others, and can be skipped in a first reading.

If the precision matrix $\boldsymbol{A}$ of the Gaussian posterior approximation $Q(\boldsymbol{u}|\boldsymbol{y})$ is sparse, the variances can be estimated in some cases using Gaussian belief propagation. For example, the algorithm in (Wainwright et al., 2001) can be used, which works by embedding a sequence of spanning trees, then does Gaussian propagation on these trees. Another interesting approach for Gaussian Markov random fields is given in (Malioutov et al., 2006a). However, $\boldsymbol{A}$ is neither sparse nor has graphical model structure[13] in general. In (Schneider and Willsky, 2001), the Lanczos algorithm (Golub and Van Loan, 1996) is used in order to estimate Gaussian posterior variances. The connection can be understood by noting that a single marginal variance can be estimated by running linear conjugate gradients (LCG) for the system $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{B}^T\boldsymbol{\delta}_i$, then $\tilde{z}_i = \boldsymbol{\delta}_i^T\boldsymbol{B}\boldsymbol{v}$. One way to regard the Lanczos algorithm is that it explicitly builds up a low rank representation of $\boldsymbol{A}$, which allows to solve *many* linear systems in parallel with the same matrix $\boldsymbol{A}$, but different right hand sides. More specifically, k iterations of Lanczos produce $\boldsymbol{Q}^{(k)} = (\boldsymbol{q}^{(1)} \dots \boldsymbol{q}^{(k)}) \in \mathbb{R}^{n\times k}$ with orthonormal columns, and $\boldsymbol{T}^{(k)} \in \mathbb{R}^{k\times k}$ tridiagonal, such that $\boldsymbol{Q}^{(k)T}\boldsymbol{A}\boldsymbol{Q}^{(k)} = \boldsymbol{T}^{(k)}$. The extremal eigenvectors of $\boldsymbol{A}$ can be well approximated in the column span of $\boldsymbol{Q}^{(k)}$ even for small k, and the convergence of such eigenvector estimates can be efficiently monitored within the Lanczos recursion. A Lanczos iteration requires a single MVM with $\boldsymbol{A}$, thus the same effort in principle than a single LCG iteration.

We obtain a Lanczos estimator for $\tilde{\boldsymbol{z}}$ by replacing $\boldsymbol{\Sigma}$ in the exact expression by $\boldsymbol{Q}^{(k)}\boldsymbol{T}^{(k)-1}\boldsymbol{Q}^{(k)T}$, resulting in $\tilde{\boldsymbol{z}}^{(k)} := \mathrm{diag}^{-1}(\boldsymbol{B}\boldsymbol{Q}^{(k)}\boldsymbol{T}^{(k)-1}\boldsymbol{Q}^{(k)T}\boldsymbol{B}^T)$. It is shown in Appendix B that $\tilde{\boldsymbol{z}}^{(k)} = \tilde{\boldsymbol{z}}^{(k-1)} + \boldsymbol{v}^{(k)} \circ \boldsymbol{v}^{(k)}$, where $\boldsymbol{v}^{(k)}$ obeys the recursion $\boldsymbol{v}^{(k)} = e_k^{-1}(\boldsymbol{B}\boldsymbol{q}^{(k)} - d_{k-1}\boldsymbol{v}^{(k-1)})$. Therefore, $\tilde{\boldsymbol{z}}^{(k)}$ converges against $\tilde{\boldsymbol{z}}$ *monotonically from below*[14] in every component. We can also estimate $\log|\boldsymbol{A}|$, featuring in the criterion part $g^*(\boldsymbol{z})$, by $\log|\boldsymbol{T}^{(k)}|$, although the latter value is not critically required in order to run our methods. While other estimators could be derived from the Lanczos method, the monotonicity of $\tilde{\boldsymbol{z}}^{(k)}$ would be lost (see end of this section). Moreover, the components of $\tilde{\boldsymbol{z}}^{(k)}$ correspond to the best estimate after k iterations of LCG for the systems $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{B}^T\boldsymbol{\delta}_i$, if the same starting vector is used for all of them.

While the basic recurrence of the Lanczos algorithm is easy to describe, this apparent simplicity is treacherous. Useful Lanczos implementations involve a lot of extra technology, and in general they require $O(n\,k)$ memory (details are given in Appendix B).

In general, a Lanczos estimate is good if the full expression dominantly depends on the extremal eigenvalues and eigenvectors, which tend to converge rapidly. An ideal case, for which most Lanczos codes seem to be optimized, is given by a matrix $\boldsymbol{A}$ with a geometrically decaying spectrum bounded away from zero. This implies a geometric decay of the spectrum of $\boldsymbol{\Sigma}$ as well, so that $\tilde{\boldsymbol{z}}^{(k)}$ converges to $\tilde{\boldsymbol{z}}$ rapidly. In applications with a geometric spectral decay of the $\boldsymbol{A}$ matrices, modern Lanczos codes, such as `ARPACK`[15], can be used. Unfortunately, for problems of our interest here (high-quality image reconstruction from non-local measurements), the system matrices $\boldsymbol{A}$ often exhibit a spectral decay which is roughly *linear* (with the possible exception of beginning and end). In such cases, $\|\tilde{\boldsymbol{z}}^{(k)} - \tilde{\boldsymbol{z}}\|/\|\tilde{\boldsymbol{z}}\|$ converges approximately linearly, and a close approximation of $\tilde{\boldsymbol{z}}$ requires $k \approx n$ iterations of Lanczos, which is not tractable. Moreover, since the Lanczos iteration delivers optimal approximations of eigenvalues and eigenvectors, given that k MVMs with $\boldsymbol{A}$ can be used, there may be no better generic method to approximate $\tilde{\boldsymbol{z}}$, if no additional knowledge about $\boldsymbol{A}$ is used. To be clear, the problem in these cases is *not* that $\boldsymbol{A}$ is ill-conditioned, but that the dependence of $\tilde{\boldsymbol{z}}$ on the eigendecomposition of $\boldsymbol{A}$ is spread across much of the spectrum. In other words, $\boldsymbol{A}$ does not have *any* low rank approximation which is close in the spectral norm. In numerical mathematics, structural knowledge about the system matrix is used in order to precondition the Lanczos algorithm, which can in principle be used to improve its spectral properties. Such preconditioning strategies are not in the scope of this paper, but remain an important topic for further investigations.

To conclude so far, we have seen that the Lanczos estimator $\tilde{\boldsymbol{z}}^{(k)}$ for $\tilde{\boldsymbol{z}}$ can be computed using k MVMs with $\boldsymbol{A}$. On the other hand, for certain system matrices $\boldsymbol{A}$ arising in practice (in problems of image reconstruction), $\tilde{\boldsymbol{z}}^{(k)}$ will not be very close to $\tilde{\boldsymbol{z}}$ in many components. This creates a problem for the global convergence proof of Section 4.4, which relies on *exact* computations of $\tilde{\boldsymbol{z}}$. In these situations, we cannot claim that our method is provably globally convergent in practice. Moreover, since the $g^*(\boldsymbol{z})$ term in $\phi_{\boldsymbol{z}}$ cannot be computed exactly, we cannot check for sure whether an outer loop step improved the criterion. From our experience on such problems,

[13] To be precise, $\boldsymbol{A}$ would be the precision matrix of the Gaussian graphical model $Q(\boldsymbol{u}|\boldsymbol{y})$, and its sparsity pattern encodes graph structure. In cases where such an interpretation applies, this structure is normally independent of the value of $\boldsymbol{\pi}$.

[14] From Theorem 1, (7.), we know that $\tilde{z}_i \in [z_i^{(k)}, \gamma_i]$ for any i and k.

[15] Available at `http://www.caam.rice.edu/software/ARPACK/`.

the algorithm is well behaved, in that the approximate $\phi_{\boldsymbol{z}}$ rapidly decreases, then jitters slightly around a value and can be stopped. Since even $\nabla_{\boldsymbol{\gamma}}\phi$ cannot be computed reliably, it is questionable whether a generic precise stopping rule can be found. Establishing such rules in particular cases of interest remains an open problem for future work.

The estimation of $\tilde{\boldsymbol{z}}$ is the most difficult computation required in our class of methods. It is done only once in each outer loop iteration, and typically only few iterations are needed at all. Recall from Section 4.1 that for a gradient descent method, $\tilde{\boldsymbol{z}}$ would have to be estimated in every single step. In Section 5.2, we compare our variational inference methods to certain MAP estimation algorithms on the same model, showing that on a purely computational level, they *mainly* differ in the computation of $\tilde{\boldsymbol{z}}$ (required for inference, but not for MAP estimation). It goes without saying that if Gaussian models with precision matrix $\boldsymbol{A}$ (under arbitrary positive $\boldsymbol{\pi}$) admit a better specific estimator for $(\mathrm{Var}_Q[s_i|\boldsymbol{y}])_i$, this should be preferred over our generic Lanczos solution here. Or the Lanczos method could be run with preconditioning, based on system knowledge. For example, if $\boldsymbol{A}$ has a sparse graphical model structure, Gaussian belief propagation on modified graphs is used to precondition LCG (Malioutov et al., 2006b), and these ideas could be used to precondition the Lanczos algorithm as well.

Surprisingly, in our experiments of main interest here, the bad relative accuracy of our estimator of $\tilde{\boldsymbol{z}}$ does *not* seem to have a major impact at all. Although the spectral decay of $\boldsymbol{A}$ is linear, and the variance estimates have significant errors, the L_2 reconstruction errors are often slightly better for smaller numbers of Lanczos iterations k than for exact computations. Here is an idea why this might be the case. Importantly, $\tilde{\boldsymbol{z}}^{(k)}$ approaches $\tilde{\boldsymbol{z}}$ componentwise from below, so we generally use *underestimates*. Moreover, the inner loop sparsity penalty for Laplace sites is $2\sum_i \tau_i(z_i + s_i^2/\sigma^2)^{1/2}$ (see Section 4.2; recall that $z_i = \tilde{z}_i$ in this case), which is stronger for smaller z_i. An underestimate of z_i leads to a stonger sparsity penalty on s_i in the subsequent inner loop, and this amplification happens mostly on those s_i, whose true z_i values would be moderately small. More sparsity is implied for $\boldsymbol{s}$ than what is specified by the prior. While this effect could be self-amplifying, it seems to be benign or even slightly beneficial in our applications, maybe because the Laplace sparsity potentials are not strong enough in the first place.

One may not be this lucky in other applications, or with other models. Indeed, it would be preferable to work with proper estimators, and adjust the model and its potentials for the right degree of sparsity. Certainly, finding better Gaussian variance estimators is an important point for future work.[16]

5 Extensions

In this section, we collect a number of extensions of the approximate inference algorithms described in Section 4. Moreover, we discuss the precise relationship to closely related sparse MAP estimation methods, pinning down the added complexity of variational inference. In this context, we also comment on compressed sensing for natural images, and on the misguided advice to measure natural or medical images with randomly drawn designs.

5.1 Direct Generalizations

It should be obvious that our framework is not limited to sparse linear models with factorizing Gaussian likelihood. It can be applied whenever the posterior can be written as the normalized product of univariate sites, whose arguments are linear combinations of $\boldsymbol{u}$, and if each of the sites has Gaussian-form lower bounds. For example, Laplace or Student's t sites can be used as *likelihood* terms for robust regression (Tipping and Lawrence, 2005). Binary classification with Bernoulli sites is discussed in Section 2.1. We can also deal with multivariate non-Gaussian sites, as long as corresponding multivariate lower bounds are given.[17] $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ become block-diagonal in such a setting. All convexity properties discussed in Section 3.1, as well as Theorem 3, remain valid if positivity requirements on scalars are replaced by requirements of positive definiteness on square blocks. The sites can have overlapping supports, since our methods can certainly deal with matrices $\boldsymbol{B}$ that do not have full rank.

We can also in principle accommodate fully coupled sites, as long as they are Gaussian. For example, the likelihood $P(\boldsymbol{y}|\boldsymbol{u})$ may come with a general covariance matrix $\boldsymbol{\Pi}^{(0)-1}$. This case is accommodated by replacing

[16] However, in the absence of such, or of a different approximate inference method as scalable as ours, the only alternative seems to not do Bayesian experimental design on a large scale at all. Fortunately, this narrow "fundamentalist" view on anything vaguely Bayesian (often ignoring real-world aspects such as running time or user-friendlyness) does not hamper work in machine learning.

[17] For example, Fenchel bounds may be generalized to multivariate scale mixtures. The latter are useful to specify correlations in non-Gaussian priors. In the context of natural images, certain filter responses in $\boldsymbol{s}$ are known to be typically correlated, and multivariate scale mixture priors have been used in this context (Portilla et al., 2003).

$\sigma^{-2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u}\|^2$ above by $(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u})^T\boldsymbol{\Pi}^{(0)}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u})$, and $\sigma^{-2}\boldsymbol{X}^T\boldsymbol{X}$ by $\boldsymbol{X}^T\boldsymbol{\Pi}^{(0)}\boldsymbol{X}$ in the system matrices. In this case, MVMs with $\boldsymbol{\Pi}^{(0)}$ (the inverse covariance matrix) have to be computed inside LCG and Lanczos iterations.

In this paper, our interest in scalable approximate Bayesian inference is mainly driven by experimental design applications to improve linear measurement architectures for images (see Section 6). Bayesian inference has many other applications, and while we do not pursue any of them in detail here, we close this section with some remarks about how our algorithms could be employed. First, the partition function $P(\boldsymbol{y})$ of Section 2.1 is an important concept on its own, besides being a favourable target for variational relaxations. It is the *marginal likelihood* of the data, where the unknown parameters $\boldsymbol{u}$ have been integrated out. Bayes factors (Kaas and Raftery, 1995) are differences of log partition functions for different models of the same data, and they are routinely used for model selection (they are the Bayesian equivalent to likelihood ratio statistics). The variational relaxation we employ results in a bound on the log partition function, and our algorithms can be used to evaluate these bounds rapidly.[18] Moreover, a powerful method for adjusting free hyperparameters, such as the τ_i scale parameters in (2), consists of maximizing the marginal likelihood $P(\boldsymbol{y})$ of the data. This is implemented easily within our variational framework, by just maximizing the lower bound to $\log P(\boldsymbol{y})$ instead. Technically, it is equivalent to maximum likelihood learning of parameters in an undirected graphical model, and the derivatives of the lower bound w.r.t. hyperparameters have the usual form of simple posterior expectations. The maximization of our bound w.r.t. the noise variance σ^2 is a convex problem if the relaxation is convex, as will be shown elsewhere. However, in general, hyperparameter optimization by marginal likelihood techniques is a non-convex problem.

5.2 Relationship to MAP Estimation. Added Complexity of Variational Inference

Our variational approximate inference methods are, from a purely computational viewpoint, closely related to MAP estimation algorithms for the same underlying posterior. A related point is made in (Wipf and Nagarajan, 2008), where they compare convex MAP with non-convex sparsity estimators. In this section, we will see that it is precisely the step from MAP estimation to variational inference which makes the outer loop updates hard. The added complexity of variational inference versus MAP estimation can be quantified precisely in this case. With this analogy in mind, we can give weight to the core messages of this paper, towards the end of this section.

The problem of *maximum a posteriori* (MAP) estimation for the sparse linear model is

$$\max_{\boldsymbol{u}} N(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{u}, \sigma^2\boldsymbol{I}) \prod_i t_i(s_i).$$

It is convex if the $t_i(s_i)$ are log-concave functions. Given that the $t_i(s_i)$ are super-Gaussian, as we assume in the rest of this section, Theorem 3 states that the variational inference approximation we employ, is convex as well. In fact, it is closely related to particular MAP estimation techniques. A key step in Section 3 is (9), where $\int(\dots)\,d\boldsymbol{u}$ is replaced by $|2\pi\sigma^2\boldsymbol{\Sigma}|^{1/2}\max_{\boldsymbol{u}}(\dots)$, an equality for Gaussian integrals. This implies that the MAP problem can be written in much the same form as the approximate inference problem, only that the $\log|\boldsymbol{A}|$ term vanishes. For the general case, the MAP estimation problem is equivalent to

$$\min_{\boldsymbol{u}}\min_{\boldsymbol{\gamma}} h(\boldsymbol{\gamma}) + \sigma^{-2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u}\|^2 + \sigma^{-2}\boldsymbol{s}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{s}, \quad h(\boldsymbol{\gamma}) = -2\sum_{i=1}^{q}\log f_i(1/\gamma_i).$$

By Theorem 3, $h(\boldsymbol{\gamma})$ is convex, therefore the whole criterion is jointly convex in $(\boldsymbol{u},\boldsymbol{\gamma})$, which allows us to interchange the ordering of $\min_{\boldsymbol{u}}$ and $\min_{\boldsymbol{\gamma}}$, and we can solve the MAP problem by iteratively updating $\boldsymbol{u}$ and $\boldsymbol{\gamma}$. This algorithm has been proposed in (Figueiredo, 2003) for the case of Laplace sites. Our Theorem 3 implies that the same algorithmic recipe can be applied to other models as well, for example (sparse) logistic regression.[19]

In general, for sites that may not be log-concave, $h(\boldsymbol{\gamma}) = h_\cap(\boldsymbol{\gamma}) + h_\cup(\boldsymbol{\gamma})$. The concave part $h_\cap(\boldsymbol{\gamma})$ is upper bounded by Fenchel's inequality, whence we repeatedly need to solve inner loop convex problems of the form

$$\min_{\boldsymbol{u}}\min_{\boldsymbol{\gamma}} \boldsymbol{z}^T\boldsymbol{\gamma} + h_\cup(\boldsymbol{\gamma}) + \sigma^{-2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{u}\|^2 + \sigma^{-2}\boldsymbol{s}^T\boldsymbol{\Pi}\boldsymbol{s}.$$

[18] Some caution is necessary here, due to the problems noted in Section 4.5.

[19] These MAP estimation algorithms may not be globally convergent for sparsity-enforcing sites, at least no proof has been given in (Figueiredo, 2003). The problem is that many γ_i have to become exactly zero eventually, since the MAP estimator for sparse linear models is provably sparse. However, if the algorithm attends some $\boldsymbol{u}$ such that $s_i = 0$ for $\boldsymbol{s} = \boldsymbol{B}\boldsymbol{u}$, then γ_i and s_i remain clamped at zero ever after. A global convergence proof for these MAP estimation methods is complete only if it is shown that only such s_i become zero, which are in fact zero at the true solution (assuming, of course, that all $s_i \neq 0$ initially). This problem does not occur for variational inference applications, since no γ_i can ever become zero there.

These inner loop steps are exactly the same as for the approximate inference method discussed in Section 4.1. However, the outer loop updates of $\boldsymbol{z}$ are different: $\boldsymbol{z}_{opt} = \nabla_{\boldsymbol{\gamma}} h_{\cap}(\boldsymbol{\gamma})$. Since $h_{\cap}$ typically decouples, these updates are simple. The $\log|\boldsymbol{A}|$ term, coupling the sites, is not present in the MAP estimation context. For example, for Student's t sites (3), we have that $\boldsymbol{z}_{opt} = (\boldsymbol{\nu} + \mathbf{1}) \circ \boldsymbol{\pi}$.

Our observations here are related to (Wipf and Nagarajan, 2008, Sect. 3). Recall from Section 4.4 that their method is useful for sparse estimation, yet does not compute the MAP estimator for the underlying model with Student's t sites, but also features a $\log|\boldsymbol{A}|$ coupling term. In their context, the coupling term helps to be very aggressively sparse. Our goal is not sparsity feature selection, but rather a faithful approximation of posterior covariances. As shown below, explicit sparsity destroys such covariance information. Since the MAP estimator for the sparse linear model exhibits exact sparsity in general, as does the method of (Wipf and Nagarajan, 2008), neither is useful for our purposes. The added complexities of obtaining meaningful uncertainty estimates required for Bayesian experimental design (see Section 6) are precisely the outer loop computations, which consist of bulk marginal variance estimations. As shown in Section 4.5, this added complexity is considerable, at least in terms of technology required. While the algorithm of (Wipf and Nagarajan, 2008) features a $\log|\boldsymbol{A}|$ term as well, its role is to *enhance* the degree of sparsity (*i.e.* , the number of $\gamma_i = 0$) beyond convex MAP estimation. The high degree of exact sparsity in $\boldsymbol{\gamma}$ helps them in turn to compute the marginal variances more efficiently (see Section 4.4).

We have seen that if $\boldsymbol{\gamma}$ becomes exactly sparse to a high degree, it is easy to implement algorithms such as ours efficiently, because the coupling matrices $\boldsymbol{A}$ are effectively only as large as the number of non-zeros in $\boldsymbol{\gamma}$. In sparse estimation, $\boldsymbol{\gamma}$ has many zeros at the final solution, and in most algorithms, $\boldsymbol{\gamma}$ is sparse from the beginning. In this setting, single-site updating algorithms (see Section 2.1) can be applicable to large problems, since single updates are cheap due to the sparsity, and the true sparse optimum may be found in a moderate number of steps. However, when the goal is approximate inference for estimating uncertainties such as posterior variances or covariances, exact sparsity of $\boldsymbol{\gamma}$ cannot be expected. By Theorem 1, (7.), the posterior variance estimate $\mathrm{Var}_Q[s_i|\boldsymbol{y}]$ is upper bounded by $\sigma^2\gamma_i$. If $\gamma_i = 0$, the method asserts that there is *no* posterior variance in s_i at all. By setting $\gamma_i = 0$, s_i is fixed to zero exactly, with absolute posterior certainty. This means that also the estimated correlation between s_i and any other s_j is zero. Since computational savings through exact sparsity can only be expected if *most* γ_i are set to zero, this means that in the corresponding posterior $Q(\boldsymbol{u}|\boldsymbol{y})$, there is no uncertainty about the large majority of the s_i, and no correlations between these and the few coefficients that survive. Basically, most coefficients are just eliminated. $Q(\boldsymbol{u}|\boldsymbol{y})$ exists on the hyperplane corresponding to the few surviving coefficients only. The very sensible and important question about *how sure* the method is in switching any of the s_i off, cannot be answered, not even a ranking among the eliminated coefficients can be extracted. In our opinion, it makes little sense to approximate Bayesian inference with such drastic side conditions, which do not come from the model or the data, but are nothing but artefacts due to the overly sparse approximation technique.

One of the main messages of our paper is that sparsity is an important statistical principle, not only for point estimation, but also if data analysis with meaningful uncertainty estimates is the goal. However, in this case, the common practice of clamping many variables to zero exactly, which is successfully used in sparse estimation, to all of our knowledge cannot be used. The goal of faithful posterior approximations has to be reached *without relying on exact sparsity*, and other large scale numerical techniques have to be used. The algorithms we use here, such as Lanczos or LCG, do not require exact sparsity in the variables to scale to very large problems. They are fast, because structure in the model matrices $\boldsymbol{X}$ and $\boldsymbol{B}$ is exploited for efficient MVMs. This structure may be sparsity (in the matrices, not in $\boldsymbol{\gamma}$ or $\boldsymbol{s}$), but does not have to be. For example, if an image is to be reconstructed from Fourier samples, MVMs with $\boldsymbol{X}$ can make use of the Fast Fourier Transform and related signal processing code. At the moment, the dominating interest in sparsity in machine learning and statistics, both in theory and in practice, seems to be to obtain ever sparser estimators, as close as possible to the most extreme case of all: L_0 regularization.[20] In some settings, such as some applications of sparse estimation, this may be the real goal (although by far the most experiments in sparse estimation papers we have seen, work with artificially generated data). But in others, certainly of at least equal importance in practice, it is not. Uncertainty estimates are basically destroyed by overly aggressive sparsification. Beyond experimental design, uncertainty estimates are acknowledged to be important in optimal decision theory. Decision-making systems based on aggressive sparse estimation may run fast, but also run the risk of getting it wrong with absolute confidence.

[20] The NP-hardness of this problem helps to understand some part of the apparent attractiveness of aggressive sparsification. But coming ever closer to a computationally hard problem with efficient relaxations does not automatically mean that real-world problems are solved better.

Compressed Sensing of Natural Images

Our experimental design method is about compressed sensing of real-world signals, and we close this section with a comment about compressed sensing and sparse estimation, which is in line with the core message of our paper. Natural images are approximately sparse in transform domains, such as wavelet or Fourier: the filter responses follow a power law, with a concentration close to zero (see Section 2). But for non-synthetic images, they are *never* exactly zero, not even in single coefficients, and certainly the dominating coefficients are not distributed uniformly at random. To see this, just transform an image, permute the wavelet coefficients randomly, and transform back: you will never retrieve anything like an image. In (Candès and Romberg, 2006, Sect. 2.2), a natural image is artificially sparsified by setting small wavelet coefficients to zero, and this latter signal (which is not a natural image) is then reconstructed from random measurements. It goes without saying that such a pre-sparsifying oracle is not something you can buy for your camera or your MR scanner: it is not realizable, and examples like this cannot tell us much about how MAP reconstruction from random measurements performs on real natural images.

Comprehensive studies on natural images can do that. The results in (Seeger and Nickisch, 2008) indicate that random measurements perform poorly on natural images, and some theoretical arguments for why this might the case, have been given in (Weiss et al., 2007). They show that for signals with a spectral decay (power as a function of spatial frequency) exhibited by natural images, the signal-to-noise ratio (SNR) of random measurements tends to zero rapidly. But the full story is likely to be even more interesting. In results to be reported elsewhere, we compare different ways of sampling 2d Fourier coefficients, which have the *same* density of samples as a function of distance from the Fourier space origin, so should give rise to the same SNR, according to the arguments of (Weiss et al., 2007). However, there is a large spread in reconstruction errors from these designs, and again, randomized designs work worst.

The problem of optimizing designs for image measurement devices is of high importance in practice, in computational photography, and even more so in medical imaging, or with cameras operating beyond the visible wavelenghts. It is a fascinating one to study, owing to the complexity of natural images, and the constraints and error sources coming with the devices. But to all of our present knowledge, it is *not solved by uniformly randomizing the measurement design*. We hope that our work here, which allows to optimize measurement designs for full images, will help along the recognition that compressed sensing is a problem about real-world signals, not about truly sparse, unstructered random vectors, and that more work than uniform random sampling is needed in order to solve it adequately.

6 Bayesian Experimental Design

In this paper, the main motivation for our scalable approximate inference method, which can maintain posteriors over entire images, is that this allows us to optimize the image measurement design $\boldsymbol{X}$ through Bayesian sequential design. This method is developed in the present section, where we also show how many large candidates for a design extension can be scored efficiently.

The framework we use here has been described in detail in (Seeger et al., 2007; Seeger, 2008), and its usefulness for optimizing image measurement architectures has been demonstrated in (Seeger and Nickisch, 2008). A clear outcome from the latter study was that while significant reductions in reconstruction error are realized by switching from linear to sparse MAP reconstruction, it is the optimization of the measurement design specifically for sparse MAP estimation that allows for much larger gains. In fact, once good designs are used, the differences in reconstruction errors between MAP and least squares reconstruction can be minor. In previous work, images of moderate size (such as 64×64) were dealt with, but the inference methods used there are not scalable. Our novel variational algorithms can be used for significantly larger images, and for the model used in these references (SLM with Laplace prior sites) solves a convex problem. In this section, we demonstrate how the design score computations can be scaled up accordingly, making use of the Lanczos algorithm once more.

6.1 Sequential Design Score Computation

In sequential experimental design[21] (Chaudhuri and Mykland, 1993; Fedorov, 1972), $\boldsymbol{X}$ is built up through several rounds. In each round, a set of candidates $\boldsymbol{X}_* \in \mathbb{R}^{d \times n}$ of equal size is scored, and the winner (maximizing the

[21] We use the term "experimental design" in a narrow sense, compared to what readers from statistics may be familiar with. We wish to quantify amounts of information in parts of experiments, and to exploit this inference to optimize the measurements automatically, where the sole aim is to obtain faithful reconstructions faster or at a lower cost. Classical ED (in our sense) concentrates on the Fisher information matrix of an estimator. For sparse MAP estimation, this matrix is not well defined, due to the shrinkage-to-zero properties (see also Section 5.2). Bayesian ED is not plagued by these problems.

score) is appended to $\boldsymbol{X}$. A candidate scores highly if its measurements are deemed to reveal as much novel information about $\boldsymbol{u}$ as possible, given what is already known at the beginning of the round. In the following, we discuss the scoring for a single round, starting from $(\boldsymbol{X}, \boldsymbol{y})$ and $P(\boldsymbol{u}|\boldsymbol{y})$. We employ the entropy difference score here,

$$\mathrm{H}[P(\boldsymbol{u}|\boldsymbol{y})] - \mathrm{E}_{P(\boldsymbol{y}_*|\boldsymbol{y})}\left[\mathrm{H}[P(\boldsymbol{u}|\boldsymbol{y}, \boldsymbol{y}_*)]\right],$$

which quantifies the reduction in posterior uncertainty (Seeger, 2008). Note that reduction in uncertainty is scored globally over all of $\boldsymbol{u}$. Importantly, the posterior correlations are fully contained in this score, setting it apart from scores based purely on marginals of $\boldsymbol{u}$. As discussed in (Seeger, 2008), its maximizer depends on the full posterior covariance matrix.

If $Q(\boldsymbol{u}|\boldsymbol{y}) = N(\boldsymbol{h}, \sigma^2\boldsymbol{\Sigma})$ is the approximation (8) to the current posterior $P(\boldsymbol{u}|\boldsymbol{y})$, we approximate this score by

$$\Delta(\boldsymbol{X}_*) := \log|\boldsymbol{\Sigma}| + \log\left|\boldsymbol{\Sigma}^{-1} + \boldsymbol{X}_*^T\boldsymbol{X}_*\right| = \log\left|\boldsymbol{I} + \boldsymbol{X}_*\boldsymbol{\Sigma}\boldsymbol{X}_*^T\right|, \tag{12}$$

the entropy difference between $Q(\boldsymbol{u}|\boldsymbol{y})$ and an updated Gaussian, where $\boldsymbol{X}_*$ is appended to $\boldsymbol{X}$, and $\boldsymbol{y}_* \sim Q(\boldsymbol{y}_*|\boldsymbol{y})$ to $\boldsymbol{y}$. This is an approximation, because we do not adapt the variational parameters $\boldsymbol{\pi}$ of $Q(\boldsymbol{u}|\boldsymbol{y})$ to the new data, but keep them at their old values. This approximation is mainly done for efficiency reasons, since our aim is to score many candidates in each round. Note that no integration over $\boldsymbol{y}_*$ is required for the $\Delta(\boldsymbol{X}_*)$ computation. Details are given in (Seeger, 2008).

Suppose there are N candidates of d rows each. The computation of (12) for these requires the solution of $N\,d$ linear systems with $\boldsymbol{A}$, but different right hand sides, which is not feasible to do with LCG in applications of our interest. We came across a related problem in Section 4.5 already, and our approach once more involves the Lanczos algorithm. Recalling the notation there, if $\boldsymbol{T}^{(k)} = \boldsymbol{L}^{(k)}\boldsymbol{L}^{(k)T}$ (Cholesky decomposition; $\boldsymbol{L}^{(k)}$ is bidiagonal) and $\tilde{\boldsymbol{Q}}^{(k)} := \boldsymbol{Q}^{(k)}\boldsymbol{L}^{(k)-T}$ (this computation is $O(n\,k)$ only), then

$$\boldsymbol{I} + \boldsymbol{X}_*\boldsymbol{\Sigma}\boldsymbol{X}_*^T \approx \boldsymbol{I} + \boldsymbol{V}_*^T\boldsymbol{V}_*, \quad \boldsymbol{V}_* := \tilde{\boldsymbol{Q}}^{(k)\,T}\boldsymbol{X}_*^T,$$

if $\boldsymbol{\Sigma}$ is replaced by its Lanczos low rank approximation. Finally, the score is computed using a Cholesky decomposition of this $d \times d$ matrix, or of $\boldsymbol{I} + \boldsymbol{V}_*\boldsymbol{V}_*^T$ (which has the same determinant) if $k < d$. In the applications we are interested in, MVMs with large $\boldsymbol{X}$ matrices are efficient, so the $\boldsymbol{V}_*$ for the different candidates are best computed en bulk, given sufficient memory. On the other hand, these score computations can directly be parallelized, given $\tilde{\boldsymbol{Q}}^{(k)}$.

The approximation described here manifests the role of the Lanczos matrices $\boldsymbol{Q}^{(k)}$, $\boldsymbol{T}^{(k)}$ of $\boldsymbol{A}$ as principal representation of the approximate posterior itself. Once these are computed, they can be used to answer rather arbitrary posterior queries, such as marginal variances of $\boldsymbol{s} = \boldsymbol{B}\boldsymbol{u}$ (in Section 4.5) or entropy difference design scores.

6.2 Other Design Setups. Relation to Classical Design

In the previous section, we showed how to compute the design score (12) for many unrelated candidates. In this section, we consider some other optimization settings for $\Delta(\boldsymbol{X}_*)$.

First, consider $d = 1$, so $\Delta(\boldsymbol{x}_*) = \log(1 + \boldsymbol{x}_*^T\boldsymbol{\Sigma}\boldsymbol{x}_*)$. As noted in (Seeger and Nickisch, 2008), the global maximizer of $\Delta(\boldsymbol{x}_*)$ among all unit-norm vectors $\boldsymbol{x}_*$ is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{\Sigma}$ (Horn and Johnson, 1985, Sect. 4.2). This makes sense intuitively, since the uncertainty in $\boldsymbol{x}_*^T\boldsymbol{u}$ is largest for this direction.[22] It should be clear now that finding optimal extensions of $\boldsymbol{X}$ with Bayesian experimental design needs an estimate of the *whole* posterior covariance matrix: its single node marginals are not enough. We can use the Lanczos algorithm (see Section 4.5) in order to find the optimal direction. In fact, finding extremal eigenvectors is typically the main application of this algorithm. Recalling the discussion of convergence properties in Section 4.5, we note that if Lanczos is applied to $\boldsymbol{A}$ with a geometrically decaying spectrum, its minimal eigenvector (the maximal eigenvector of $\boldsymbol{\Sigma} = \boldsymbol{A}^{-1}$) comes out last. In such a situation, packages like ARPACK run Lanczos on $\boldsymbol{A}^{-1}$ instead, which requires LCG in each iteration. However, for applications of our interest here, the linear spectral decay of $\boldsymbol{A}$ means that eigenvectors from both ends of the spectrum converge rapidly, so we can just run Lanczos on $\boldsymbol{A}$, until the minimal eigenvector converges.

[22] However, our simplifying assumption leading to (12) is also witnessed here, in that the outcome of a measurement $\boldsymbol{x}_*$ is assumed to mainly reduce the uncertainty in $\boldsymbol{x}_*^T\boldsymbol{u}$ alone, while any further non-Gaussian "spread" of uncertainty reduction (which could happen, if $\boldsymbol{\pi}$ was updated for the score computation) is ignored.

It is also possible to compute $\partial\Delta(\boldsymbol{X}_*)/(\partial\alpha)$ if $\partial\boldsymbol{X}_*/(\partial\alpha)$ is known. If $\boldsymbol{M} = \boldsymbol{I} + \boldsymbol{X}_*\boldsymbol{\Sigma}\boldsymbol{X}_*^T$ and $\boldsymbol{W}_* = \tilde{\boldsymbol{Q}}^{(k)\,T}(\partial\boldsymbol{X}_*/(\partial\alpha))^T$, then

$$\partial\Delta(\boldsymbol{X}_*)/(\partial\alpha) = \mathrm{tr}\,\boldsymbol{M}^{-1}\left(\boldsymbol{W}_*^T\boldsymbol{V}_* + \boldsymbol{V}_*^T\boldsymbol{W}_*\right).$$

If $k < d$, we can work with $\boldsymbol{I} + \boldsymbol{V}_*\boldsymbol{V}_*^T$ instead.

How difficult is the optimization of $\Delta(\boldsymbol{X}_*)$ in general, over infinitely many candidates $\boldsymbol{X}_*$? For the case $d = 1$, we have to maximize the convex quadratic $\boldsymbol{x}_*^T\boldsymbol{\Sigma}\boldsymbol{x}_*$, which is easy if the feasible set of $\boldsymbol{x}_*$ is a Euclidean ball or ellipse: the solution is a generalized eigenproblem. However, in general, convex quadratic *maximization* is a hard problem, even for linear constraints on $\boldsymbol{x}_*$. This is not surprising, since finding the optimal design in a Gaussian linear model is already a hard problem in principle, at least for large n and q.

From a purely computational viewpoint, our optimization of $\Delta(\boldsymbol{X}_*)$ is related to classical sequential D-optimal design, or its Bayesian analogue for Gaussian linear models (Chaloner and Verdinelli, 1995), with the important difference that the Fisher information matrix (which would be $\boldsymbol{X}^T\boldsymbol{X}$ for the Gaussian linear model) is replaced by the posterior precision matrix $\boldsymbol{A}$ here. Therefore, the work referenced in (Chaudhuri and Mykland, 1993) can be used in our framework as well. In the nomenclature of (Chaloner and Verdinelli, 1995), our sparse Bayesian design framework is *nonlinear*. While they note that $P(\boldsymbol{u}|\boldsymbol{y})$ is often approximated by a Gaussian, none of the methods they refer to uses a modern variational approximation. With smooth non-Gaussian sites, the Laplace approximation is typically employed in statistics to obtain a Gaussian posterior approximation. However, in the case of sparse linear models, the log posterior is strongly singular at its mode, so that the Laplace approximation is not well defined. Moreover, the fact $m \ll n$ invalidates the typical justification for this approximation, as well as all asymptotic results about it we know of. Classical D-optimal design, as well as its Bayesian variant for Gaussian linear models, are *linear* techniques, meaning that the design score to optimize does not depend on the observations $\boldsymbol{y}$. The design optimization is done without ever looking at any real data. In the applications of our interest, while the linearity of the measurements and the sparsity properties of the signal $\boldsymbol{u}$ can be motivated well, the model setup certainly does not perfectly represent the true data-generating process. The dependence of design decisions on data gathered along the process is an important feature of the our method, rendering it robust against model mismatch, which is surely present. Nonlinear sequential design based on maximum likelihood estimation has been analyzed in (Chaudhuri and Mykland, 1993), and an interesting point for future research would be to extend their analysis to settings such as ours, where modern variational approximations of the posterior covariance matrix are used in place of the inverse Fisher information matrix, and MAP or posterior mean estimators replace maximum likelihood techniques.

7 Discussion

Many modern applications of statistical inference and estimation come with a large number of latent variables, often many more than the number of independent datapoints. While a recent surge of activity has established efficient convex methods for sparse point estimation from such data, little work has been done on higher-order problems for sparsity-favouring models, such as estimating confidences and dependencies, or optimizing measurement architectures. These problems can be addressed by Bayesian inference, but the commonly used standard approaches, such as Laplace approximation or Markov chain Monte Carlo, either do not apply (the Hessian is strongly singular at the mode; see also Section 5.2) or do not run fast enough for many real-world setups. Variational approximations to Bayesian inference have been applied to sparse linear models (Tipping, 2001; Girolami, 2001; Figueiredo, 2003; Wipf et al., 2004), but the algorithms known so far scale up to large problems only if the main objective is sparse *estimation* once more. Moreover, the variational relaxations as well as the algorithms for solving them, have not been given satisfying characterizations.

In this paper, we concentrate on a widely used general variational relaxation based on Gaussian-form lower bounds (Jaakkola, 1997; Palmer et al., 2006; Girolami, 2001), which for a range of models is equivalent to the variational (mean field) Bayes technique. We settle a long-standing question for this approximation, by showing that the variational problem is a convex minimization if and only if the search for the posterior mode is. Moreover, and probably of more importance in practice, we provide the first truly scalable algorithms for solving this minimization (whether convex or not) on large scale models. Scalability is achieved, just as in many convex sparse estimation algorithms, by reducing the dominating efforts to problems of standard form, with very well developed solutions in numerical mathematics: least squares estimation, and variance estimation in Gaussian random fields. If code for these primitives is in place, the implementation of our methods is straightforward. Our setup is generic and can be configured with little effort to models featuring sparsity potentials (Laplace, Student's t; see (Palmer

et al., 2006) for further super-Gaussian potentials), binary classification likelihoods, or many other exponential family sites. Since the dominating computations are spent in standard primitives, no additional heuristics have to be tuned. Moreover, structure in the model coupling matrices (measurement design, prior filters) can be exploited very effectively.

Our main interest in this paper is Bayesian experimental design, with the aim of optimizing measurement architectures for natural images. Uniform random sampling is not enough to find useful measurement designs for real-world signals (this point is discussed in more detail at the end of Section 5.2). The ability of estimating posterior covariances is crucial for finding good designs, and methods for aggressive sparse estimation cannot in general be used towards this end (they are important for reconstruction, once a good design has been found). Moreover, the queries required to improve a design can be approximated using the same primitives our inference algorithms rely on.

Our algorithms are of the double loop, or difference of convex type, which holds much promise for many problems in machine learning. The idea is to decouple a criterion ϕ, for which even the gradient is hard to compute, by upper bounding a critical part, so that the resulting bound $\phi_{\boldsymbol{z}}$ can be minimized efficiently. This decoupling is successful if much of the criterion structure is still contained in the bound, so that only a few outer loop steps (re-fits of the bound) are required until convergence. This may not happen for all instances of our algorithms. If strong couplings are eliminated by the bounding step, the inner loop optimization tends to converge rapidly without much progress in ϕ, and many outer loop steps are required. Since double loop algorithms are increasingly used in machine learning and statistics (EM algorithm, CCCP, variational Bayes), it is an important point for further research to understand under which conditions they work well, and what can be done if they do not.[23]

The algorithms proposed here have been shown to work well on challenging large scale medical imaging problems, but are certainly useful in many other applications as well. In time series, filtering, or tracking problems, large state spaces could be endowed with sparsity priors, involving potentials between subsequent time points. Systems biology applications create datasets with very many latent variables, and experimental design can be used there to save on expensive experiments (Steinke et al., 2007). Sparsity models can also be used to analyze data from neural cell recordings, allowing for sharper predictions than traditional second order correlation analysis (Gerwinn et al., 2008). Our methods should be especially interesting for low-level computer vision applications, since sparsity-enforcing models represent natural image statistics much better than purely Gaussian ones, yet our methods draw exclusively on primitives such as mean and variance estimation in *Gaussian* Markov random fields, for which recently very efficient algorithms have been proposed (Malioutov et al., 2006a,b). In this sense, our algorithms reduce inference for certain non-Gaussian MRFs to (repeated) computations in Gaussian MRFs. The SLMs we employ in this paper, are already used for problems such as image denoising or super-resolution, by way of sparse estimation. Beyond these applications close to machine learning, our techniques can be imported rather easily into application fields, where least squares estimation or lower-level signal processing techniques are dominatingly used, simply because our methods are configured solely in terms of such well-studied computational primitives.

Finally, we hope that our work here contributes to the wider recognition of scalable Bayesian approximations in computational application fields. It should become clear from our results that maximum a posteriori (MAP) estimation is not Bayesian inference (see Section 5.2 for a discussion), although even in machine learning or computer vision, Bayesian techniques are often equated with MAP throughout. Bayesian inference is about integration, while MAP is about optimization towards a single point estimate. While MAP estimates of the unknowns themselves can often be computed by convex programming, covariances between variables or other uncertainty queries cannot properly be estimated this way. Variational approximations reduce Bayesian integration to optimization problems, and although by far the most work in this field is done on discrete graphical models, similar principles apply to continuous variable models as well. In fact, our work shows that resulting algorithms can fit in seemlessly, running on the same computational primitives than MAP or least squares estimation, and thereby deal with long-range couplings and strongly non-local measurements in a way that is not currently possible for discrete random fields methods.

[23] Some hints may come from loopy belief propagation for discrete graphical models, where double loop methods like CCCP (Yuille and Rangarajan, 2003) have been proposed, yet are much too slow to be practical.

Acknowledgments

We thank Rolf Pohmann, Bernhard Schölkopf, Florian Steinke, and David Wipf for helpful discussions and input. Supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

A Further Details

In this section, we collect details of arguments which have been ommitted in the text.

A.1 Proof of Theorem 1

In this section, we provide the proof of Theorem 1. We begin with (5.). Define $\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\rho(\boldsymbol{\Gamma})\boldsymbol{B}$ for now, where $\rho_i(\gamma_i) > 0$ for all i, and $\psi_1 = \log|\boldsymbol{A}|$. We have that $d\psi_1 = \operatorname{tr}\boldsymbol{S}\boldsymbol{D}$ with $\boldsymbol{S} = \boldsymbol{B}\boldsymbol{A}^{-1}\boldsymbol{B}^T$ and $\boldsymbol{D} = \rho'(\boldsymbol{\Gamma})(d\boldsymbol{\Gamma})$. Now, since $d\boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}(d\boldsymbol{A})\boldsymbol{A}^{-1}$, we have that

$$d^2\psi_1 = -\operatorname{tr}\boldsymbol{S}\boldsymbol{D}\boldsymbol{S}\boldsymbol{D} + \operatorname{tr}\boldsymbol{S}\rho''(\boldsymbol{\Gamma})(d\boldsymbol{\Gamma})^2 = \operatorname{tr}\boldsymbol{D}\boldsymbol{S}\boldsymbol{D}(g(\boldsymbol{\Gamma}) - \boldsymbol{S}),$$

where $g_i(\gamma_i) = \rho_i''(\gamma_i)/(\rho_i'(\gamma_i))^2$. Since $\boldsymbol{S} \succeq \boldsymbol{0}$ (positive semi-definite) we can write $\boldsymbol{S} = \boldsymbol{V}\boldsymbol{V}^T$ with some matrix $\boldsymbol{V}$, and $d^2\psi_1 = \operatorname{tr}(\boldsymbol{D}\boldsymbol{V})^T(g(\boldsymbol{\Gamma}) - \boldsymbol{S})\boldsymbol{D}\boldsymbol{V}$. If we can show that $g(\boldsymbol{\Gamma}) - \boldsymbol{S} \succeq \boldsymbol{0}$, then for $\boldsymbol{\gamma}^{(t)} = \boldsymbol{\gamma} + t(\Delta\boldsymbol{\gamma})$, we have that $\psi_1''(0) = \operatorname{tr}\boldsymbol{M}^T(g(\boldsymbol{\Gamma}) - \boldsymbol{S})\boldsymbol{M} \geq 0$ for all small $\Delta\boldsymbol{\gamma}$, where $\boldsymbol{M} = \rho'(\boldsymbol{\Gamma})(\Delta\boldsymbol{\Gamma})\boldsymbol{V}$. This implies convexity of ψ_1.

Next, we show that $\rho(\boldsymbol{\Gamma})^{-1} - \boldsymbol{S} \succeq \boldsymbol{0}$. Our proof employs the identity

$$\boldsymbol{r}^T\boldsymbol{M}^{-1}\boldsymbol{r} = \max_{\boldsymbol{x}} 2\boldsymbol{r}^T\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x}, \tag{13}$$

which holds whenever $\boldsymbol{M} \succ \boldsymbol{0}$ (positive definite). For any vector $\boldsymbol{r} \in \mathbb{R}^q$, we have that

$$\boldsymbol{r}^T\boldsymbol{B}\boldsymbol{A}^{-1}\boldsymbol{B}^T\boldsymbol{r} = \max_{\boldsymbol{x}} 2\boldsymbol{r}^T\boldsymbol{B}\boldsymbol{x} - \boldsymbol{x}^T\left(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\rho(\boldsymbol{\Gamma})\boldsymbol{B}\right)\boldsymbol{x} \leq \max_{\boldsymbol{k}=\boldsymbol{B}\boldsymbol{x}} 2\boldsymbol{r}^T\boldsymbol{k} - \boldsymbol{k}^T\rho(\boldsymbol{\Gamma})\boldsymbol{k},$$

using (13) and $\boldsymbol{x}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{x} = \|\boldsymbol{X}\boldsymbol{x}\|^2 \geq 0$. Now, if the maximum is taken over *all* $\boldsymbol{k} \in \mathbb{R}^q$, the expression cannot become smaller, so

$$\boldsymbol{r}^T\boldsymbol{S}\boldsymbol{r} \leq \max_{\boldsymbol{k}} 2\boldsymbol{r}^T\boldsymbol{k} - \boldsymbol{k}^T\rho(\boldsymbol{\Gamma})\boldsymbol{k} = \boldsymbol{r}^T\rho(\boldsymbol{\Gamma})^{-1}\boldsymbol{r},$$

using (13) once more. Therefore, $\rho(\boldsymbol{\Gamma})^{-1} - \boldsymbol{S} \succeq \boldsymbol{0}$. Note that this argument, applied to $\rho_i(\gamma_i) = \gamma_i^{-1}$ and $\boldsymbol{r} = \boldsymbol{\delta}_i$, proves (7.).

Collecting all parts, the convexity of $\boldsymbol{\gamma} \mapsto \psi_1$ is implied by $g(\boldsymbol{\Gamma}) - \rho(\boldsymbol{\Gamma})^{-1} \succeq \boldsymbol{0}$. An elementary computation shows that the latter is implied by $\rho_i(\gamma_i)\rho_i''(\gamma_i) \geq (\rho_i'(\gamma_i))^2$ for all γ_i. This completes the proof of (5.).

We continue with (6.). Define $\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}^T\rho(\boldsymbol{\Gamma})^{-1}\boldsymbol{B}$, and $\psi_2 = \log|\boldsymbol{A}| + \log|\rho(\boldsymbol{\Gamma})|$. The concavity of $\boldsymbol{\gamma} \mapsto \psi_2$ is shown by induction on q, the number of rows of $\boldsymbol{B}$. Assume for now that $\boldsymbol{X}^T\boldsymbol{X}$ is nonsingular. First, let $q = 1$, and $\boldsymbol{b} = \boldsymbol{B}^T$. Then,

$$\log\rho_1(\gamma_1) + \log\left|\boldsymbol{X}^T\boldsymbol{X} + \rho_1(\gamma_1)^{-1}\boldsymbol{b}\boldsymbol{b}^T\right| = \log\left|\boldsymbol{X}^T\boldsymbol{X}\right| + \log\left(\rho_1(\gamma_1) + \boldsymbol{b}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{b}\right).$$

Now, $\log(\cdot)$ is concave and nondecreasing, so the concavity for $q = 1$ follows from (Boyd and Vandenberghe, 2002, Sect. 3.2.4). If $q > 1$, let $\boldsymbol{B} = (\boldsymbol{B}_{<q}^T\,\boldsymbol{b})^T$ and $\boldsymbol{A}_{<q} = \boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{B}_{<q}^T\rho(\boldsymbol{\Gamma}_{<q})^{-1}\boldsymbol{B}_{<q}$. Then,

$$\psi_2 = \log|\rho(\boldsymbol{\Gamma})| + \log\left|\boldsymbol{A}_{<q} + \rho_q(\gamma_q)^{-1}\boldsymbol{b}\boldsymbol{b}^T\right| = \log|\rho(\boldsymbol{\Gamma}_{<q})| + \log|\boldsymbol{A}_{<q}| + \log\left(\rho_q(\gamma_q) + \boldsymbol{b}^T\boldsymbol{A}_{<q}^{-1}\boldsymbol{b}\right).$$

The sum of the first two terms is concave by assumption. Since $\log(\cdot)$ is concave and nondecreasing, the concavity of the final term follows from the concavity of $\boldsymbol{\gamma} \mapsto \boldsymbol{b}^T\boldsymbol{A}_{<q}^{-1}\boldsymbol{b}$ (Boyd and Vandenberghe, 2002, Sect. 3.2.4). Using (13), we have

$$\boldsymbol{b}^T\boldsymbol{A}_{<q}^{-1}\boldsymbol{b} = \max_{\boldsymbol{x}} 2\boldsymbol{b}^T\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{A}_{<q}\boldsymbol{x} = \max_{\boldsymbol{x}} Q(\boldsymbol{x}) - \boldsymbol{v}^T\rho(\boldsymbol{\Gamma}_{<q})^{-1}\boldsymbol{v},$$

with $\boldsymbol{v} = \boldsymbol{B}_{<q}\boldsymbol{x}$ and $Q(\boldsymbol{x})$ a concave quadratic. If $\boldsymbol{\rho} = \rho(\boldsymbol{\gamma}_{<q})$, the right hand side argument of $\max_{\boldsymbol{x}}$ is jointly concave as a function of $(\boldsymbol{x}, \boldsymbol{\rho})$, $\boldsymbol{\rho} \succ \boldsymbol{0}$ (negative *quadratic-over-linear*, see Section 3.1), so that $\boldsymbol{\rho} \mapsto \boldsymbol{b}^T\boldsymbol{A}_{<q}^{-1}\boldsymbol{b} =: \kappa(\boldsymbol{\rho})$ is concave for $\boldsymbol{\rho} \succ \boldsymbol{0}$ (Boyd and Vandenberghe, 2002, Sect. 3.2.5). Moreover, for any $i \in \{1, \ldots, q\}$ and any $\Delta > 0$,

$$\kappa(\boldsymbol{\rho} + \Delta\boldsymbol{\delta}_i) = \boldsymbol{b}^T\left(\boldsymbol{A}_{<q} - \frac{\Delta}{\rho_i(\rho_i + \Delta)}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{b} \geq \boldsymbol{b}^T\boldsymbol{A}_{<q}^{-1}\boldsymbol{b} = \kappa(\boldsymbol{\rho}),$$

so κ is nondecreasing in each of its arguments, and the concavity of $\boldsymbol{\gamma} \mapsto \boldsymbol{b}^T \boldsymbol{A}_{<q}^{-1} \boldsymbol{b}$ follows from (Boyd and Vandenberghe, 2002, Sect. 3.2.4). This concludes the proof, under the assumption that $\boldsymbol{X}^T \boldsymbol{X}$ is invertible.

If $\boldsymbol{X}^T \boldsymbol{X}$ is singular, define ψ_2^ε as above, but with $\boldsymbol{X}^T \boldsymbol{X} \to \boldsymbol{X}^T \boldsymbol{X} + \varepsilon \boldsymbol{I}$. ψ_2^ε is concave for all $\varepsilon > 0$. For any $\boldsymbol{\gamma} \succ \boldsymbol{0}$ s.t. $\psi_2(\boldsymbol{\gamma}) > -\infty$, ψ_2^ε converges uniformly to ψ_2 on a closed environment of $\boldsymbol{\gamma}$ (ψ_2 and all ψ_2^ε are continuous), so that ψ_2 is concave around $\boldsymbol{\gamma}$. This completes the proof of (6.).

A.2 Proof of Theorem 3

In this section, we prove Theorem 3. In fact, by Theorem 2, we only need to establish the convexity of $h(\boldsymbol{\gamma}) = -2 \sum_i \log f_i(1/\gamma_i)$. Recall super-Gaussianity from Section 2.1. To simplify notation, we ignore some positive scaling, they do not alter convexity properties. We also ignore additional linear terms $\alpha_i s_i$ in $\log t_i(s_i)$, since they can be dealt with as noted at the end of Section 2.1, leading to an additional linear term in $\phi(\boldsymbol{\gamma})$. In the following, we pick an index $i \in \{1, \dots, q\}$, and drop the corresponding subscript.

Let $x = s^2$ and $g(s) = g(x) = \log t(s)$. $g(s)$ is odd, and we only deal with $s \geq 0$ in the following. $s \mapsto g(s)$ is concave and twice continuously differentiable for $s > 0$, and $x \mapsto g(x)$ is strictly convex and nonincreasing for $x > 0$. By Fenchel duality,

$$h(\gamma) = g^*(-1/\gamma) = \sup_{s \geq 0} f = \sup_{x \geq 0} f, \quad f = -s^2/\gamma - g(s) = -x/\gamma - g(x).$$

We start with a simple, general observation. Let $0 < \gamma < \gamma'$, so that $-g(0) < h(\gamma), h(\gamma') < \infty$. If $s_* := \operatorname{argmax}_s f(s, \gamma)$ is a maximum point,[24] then $s_* > 0$, and $h(\gamma') \geq -s_*^2/\gamma' - g(s_*) > -s_*^2/\gamma - g(s_*) = h(\gamma)$. Therefore, if $\gamma_0 = \sup\{\gamma \,|\, f(s, \gamma) \leq -g(0) \ \forall s\}$ ($\gamma_0 = 0$ if this set is empty), then $s_* = 0$, $h(\gamma) = -g(0)$ for $0 < \gamma \leq \gamma_0$, and for $\gamma > \gamma_0$, h is strictly increasing, and $s_* > 0$ (note that $s_* = \infty$ is allowed). Therefore, it suffices to show that h is convex at all γ where $s_* \in (0, \infty)$.

Here and in the following, $g' = dg/(ds)$, *etc*. We use the notation $f_s = \partial f/(\partial s)$, functions are evaluated at (s_*, γ) if nothing else is said. Now, $f_s = -2s_*/\gamma - g'(s_*) = 0$, so that

$$g'(s_*) = -2s_*/\gamma. \tag{14}$$

Next, $x \mapsto g(x)$ is twice continuously differentiable as well. We have $x_* = s_*^2$ at γ. f_x is continuously differentiable, and $g''(x) > 0$ by the strict convexity of $g(x)$. By the implicit function theorem, $x_*(\cdot)$ is continuously differentiable at γ, and since $h(\gamma) = f(x_*(\gamma), \gamma)$, $h'(\gamma)$ exists. Moreover, $0 = (d/d\gamma) f_x(x_*(\gamma), \gamma) = f_{x,\gamma} + f_{x,x}(dx_*)/(d\gamma)$, so that $(dx_*)/(d\gamma) = \gamma^{-2}/g''(x_*) > 0$, and x_* is increasing in γ.

From $f_s = 0$, we have that $h'(\gamma) = f_\gamma = s_*^2/\gamma^2 = (g'(s_*))^2/4$ by (14). Now, $g'(s)$ is nonincreasing by the concavity of $g(s)$, and $g'(s_*) < 0$ by (14), which means that $s_* \mapsto h'(\gamma)$ is nondecreasing. Since s_*^2 is increasing in γ, so is s_*. Therefore, $\gamma \mapsto h'(\gamma)$ is nondecreasing, thus $h(\gamma)$ is convex for $s_* \in (0, \infty)$.

It remains to show necessity of the concavity of $g(s)$. Suppose that $g''(\tilde{s}) > 0$ for some $\tilde{s} > 0$. From (14) we see that $g'(s_*) < 0$ whenever $s_* > 0$, and by the same equation, there exists a $\tilde{\gamma} > 0$ so that $s_*(\tilde{\gamma}) = \tilde{s}$. But if $g''(s_*) > 0$ at $\tilde{\gamma}$, then $s_* \mapsto h'(\gamma)$ is decreasing at $s_* = \tilde{s}$, and just as above $h'(\gamma)$ is decreasing at $\tilde{\gamma}$, so that $h(\gamma)$ is not convex. Reverting to subscripts, suppose that h_i is not convex at $\tilde{\gamma}_i > 0$. Setting $\boldsymbol{y} = \boldsymbol{0}$ in (10) leaves us with $\phi = h(\gamma_i) + \log |\boldsymbol{A}|$, viewed as a function of γ_i. We can easily construct a setup so that $\partial^2 \log |\boldsymbol{A}|/(\partial \gamma_i^2) < -h_i''(\tilde{\gamma}_i)/2$ at $\gamma_i = \tilde{\gamma}_i$, showing that ϕ is not in general convex.

A.3 Generic Inner Loop Criterion

Recall the inner loop criterion (11), and its minimization by IRLS (see Section 4.3). In this section, we show how IRLS can be run if $h_{\cup,i}^*(\gamma_i)$ cannot be determined analytically. In this case, its value and θ_i, ρ_i have to be computed on demand for each s_i encountered. In the following, we fix $i \in \{1, \dots, q\}$ and drop the subscript.

We only treat first class inner loops, the procedure for the second class is very similar. Let $k(s, \gamma) = z\gamma + h_\cup(\gamma) + (s^2/\sigma^2)\gamma^{-1}$. Then $h_\cup^*(\gamma) = k(s, \gamma_*)$, where the minimizer γ_* is found by univariate convex minimization. Since $k_\gamma(s, \gamma_*) = 0$, we have that $\theta = (\sigma^2/2)(2/\sigma^2) s/\gamma_* = s/\gamma_*$. Next, because $k_\gamma(s, \gamma_*) = 0$ for all s, then $0 = (d/ds) k_\gamma(s, \gamma_*) = k_{s,\gamma} + k_{\gamma,\gamma} \cdot (d\gamma_*)/(ds)$ (always evaluated at s, γ_*), leading to

$$\frac{d\gamma_*}{ds} = \frac{s\gamma_*}{s^2 + \gamma_* \kappa}, \quad \kappa = \frac{\sigma^2}{2} \gamma_*^2 h_\cup''(\gamma_*). \tag{15}$$

[24] We do not require that s_* is unique ($f(s_*, \gamma)$ is unique, since $x \mapsto f(x, \gamma)$ is concave), but only that $f(s_*, \gamma) > f(0, \gamma)$. If the supremum is not attained at any point, then $s_* = \infty$ (by the continuity of f, this happens only if $\max_{s \in [0, s_0]} f(s, \gamma) < h(\gamma)$ for all s_0).

Finally, $\rho = (d\theta)/(ds) = (\gamma_*)^{-1}[1 - \theta(d\gamma_*)/(ds)] = \kappa/(s^2 + \gamma_*\kappa)$.

Moreover, not even $h_\cup(\gamma)$ needs to be known analytically. We show how to deal with the case $h_\cup(\gamma) = h(\gamma)$, other cases being similar. All we need is code in order to compute $g(x)$ and its first and second derivative. Recall that $g(x)$ is convex, and $h(\gamma) = -\min_{x\ge 0} l(x,\gamma)$, with $l(x,\gamma) = x/\gamma + 2g(x)$. The procedure requires the computation of $h(\gamma)$, $h'(\gamma)$, and $h''(\gamma)$ on demand. First, $h(\gamma) = -l(x_*,\gamma)$, where x_* is found by univariate convex minimization. Since $l_x = 0$ (all functions evaluated at (x_*,γ)), we have that $h'(\gamma) = -l_\gamma = x_*/\gamma^2$. Moreover, $(d/d\gamma)l_x(x_*(\gamma),\gamma) = 0$, so that $(dx_*)/(d\gamma) = \gamma^{-2}/(2g''(x_*))$ (recall that $g''(x_*) > 0$). Therefore, $h''(\gamma) = \gamma^{-4}((2g''(x_*))^{-1} - 2x_*\gamma)$. The second derivative is needed in (15) only, we have that

$$\kappa = \gamma_*^{-1}\sigma^2\left(\frac{1}{4\gamma_* g''(x_*)} - x_*\right),$$

where $x_* = x_*(\gamma_*)$.

A.4 Two-Parameter Gaussian Site Bounds

If $t_i(s_i)$ is not even, the Gaussian-form lower bound may not be centered at zero, as happens in the case of Bernoulli potentials (7). More generally, the Gaussian form may come with a second parameter $\tilde{b}_i$ controlling its position. In this case, $Q(\boldsymbol{u}) = C_2^{-1}N^U(\boldsymbol{s}|\sigma^{-2}\tilde{\boldsymbol{b}}, \sigma^{-2}\boldsymbol{\Pi})$. Our criterion ϕ remains as in (10), but $\boldsymbol{s}^T\boldsymbol{\Pi}\boldsymbol{s}$ is replaced by $\boldsymbol{s}^T\boldsymbol{\Pi}\boldsymbol{s} - 2\tilde{\boldsymbol{b}}^T\boldsymbol{s}$. Let $\boldsymbol{b} := \boldsymbol{\Pi}^{-1}\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{s}} := \boldsymbol{B}\boldsymbol{u} - \boldsymbol{b}$. Then,

$$\phi(\boldsymbol{\gamma},\boldsymbol{b}) = \log|\boldsymbol{A}| - \sigma^{-2}\boldsymbol{b}^T\boldsymbol{\Pi}\boldsymbol{b} + h(\boldsymbol{\gamma},\boldsymbol{b}) + \sigma^{-2}\min_{\boldsymbol{u}}\left(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{u}\|^2 + \tilde{\boldsymbol{s}}^T\boldsymbol{\Pi}\tilde{\boldsymbol{s}}\right).$$

Here, $(\boldsymbol{u},\boldsymbol{\gamma},\boldsymbol{b}) \mapsto \tilde{\boldsymbol{s}}^T\boldsymbol{\Pi}\tilde{\boldsymbol{s}}$ is jointly convex, giving rise to an inner loop much as in Section 4.3. Moreover, $(\boldsymbol{\gamma},\boldsymbol{b}) \mapsto -\boldsymbol{b}^T\boldsymbol{\Pi}\boldsymbol{b}$ is jointly concave, so that either of our classes of algorithms can be generalized (using bounds linear in $\boldsymbol{b}$ and $\boldsymbol{\gamma}$).

B The Lanczos Algorithm

The Lanczos algorithm (Golub and Van Loan, 1996) is a standard tool of numerical mathematics. For a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is positive definite, the linear conjugate gradients (LCG) algorithm (Golub and Van Loan, 1996) is a method to approximate $\boldsymbol{x}_* = \boldsymbol{A}^{-1}\boldsymbol{c}$, by minimizing the convex quadratic $q(\boldsymbol{x}) := 2\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$. In each iteration, a single matrix-vector multiplication (MVM) with $\boldsymbol{A}$ is required. The sequence of Krylov subspaces is defined as $\mathcal{K}_k := \mathrm{span}\{\boldsymbol{c}, \boldsymbol{A}\boldsymbol{c}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{c}\}$. The outcome of LCG after k iterations is the minimizer of $q(\boldsymbol{x})$ within $\mathcal{K}_k$. The Lanczos algorithm is an extension of LCG. In iteration k, a unit norm vector $\boldsymbol{q}^{(k+1)}$ is generated, which is orthogonal to all previous $\boldsymbol{q}^{(j)}$, $j \le k$, so that $\mathcal{K}_k = \mathrm{span}\{\boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(k)}\}$. In the Lanczos sequence, $\boldsymbol{q}^{(k+1)}$ is generated by a recurrence involving $\boldsymbol{q}^{(k)}$ and $\boldsymbol{q}^{(k-1)}$ only, and still only a single MVM with $\boldsymbol{A}$ per iteration is required. On finite-precision computers, a number of additional techniques are required to obtain a useful algorithm (see Appendix B.1). In general, the quantities of interest to be estimated by the Lanczos method do not depend on the right hand side $\boldsymbol{c}$. We follow the custom of using randomly drawn unit norm vectors $\boldsymbol{c}$.

The Lanczos method constructs an orthonormal $\boldsymbol{Q}^{(k)} := (\boldsymbol{q}^{(1)} \ldots \boldsymbol{q}^{(k)})$ and a tridiagonal $\boldsymbol{T}^{(k)}$ with main diagonal $\boldsymbol{\alpha} \in \mathbb{R}^k$ and subdiagonal $\boldsymbol{\beta} \in \mathbb{R}^{k-1}$, so that $\boldsymbol{Q}^{(k)T}\boldsymbol{A}\boldsymbol{Q}^{(k)} = \boldsymbol{T}^{(k)}$. Here, leading eigenvalues of $\boldsymbol{A}$ are close to leading eigenvalues of $\boldsymbol{T}^{(k)}$ rapidly, and this convergence of eigenvalues and eigenvectors can be tested within the algorithm itself (Parlett and Scott, 1979). The rough idea behind Lanczos estimates of linear functions of $\boldsymbol{A}$ or $\boldsymbol{\Sigma} = \boldsymbol{A}^{-1}$ is to plug in the following low rank approximations

$$\boldsymbol{A} \mapsto \boldsymbol{Q}^{(k)}\boldsymbol{T}^{(k)}\boldsymbol{Q}^{(k)T}, \quad \boldsymbol{\Sigma} \mapsto \boldsymbol{Q}^{(k)}\boldsymbol{T}^{(k)-1}\boldsymbol{Q}^{(k)T}.$$

If the function of interest depends on the spectrum of $\boldsymbol{A}$ only, the spectrum of $\boldsymbol{T}^{(k)}$ is used instead. For example, the Lanczos estimate of $\tilde{\boldsymbol{z}} = \mathrm{diag}^{-1}(\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T)$ in Section 4.5 is given by

$$\tilde{\boldsymbol{z}}^{(k)} := \mathrm{diag}^{-1}\left(\boldsymbol{B}\boldsymbol{Q}^{(k)}\boldsymbol{T}^{(k)-1}\boldsymbol{Q}^{(k)T}\boldsymbol{B}^T\right).$$

Since the $\boldsymbol{T}^{(k)}$ are tridiagonal and nested, it is easy to derive a recurrence for these estimates. $\boldsymbol{T}^{(k)}$ is positive definite just as $\boldsymbol{A}$. Let $\boldsymbol{T}^{(k)} = \boldsymbol{L}^{(k)}\boldsymbol{L}^{(k)T}$ be its Cholesky decomposition, where $\boldsymbol{L}^{(k)}$ is lower triangular. The

factors are bidiagonal, say $\boldsymbol{e} := \mathrm{diag}^{-1}(\boldsymbol{L}^{(k)})$, and $\boldsymbol{d} \in \mathbb{R}^{k-1}$ the subdiagonal. The following recurrence takes us from $\boldsymbol{L}^{(k-1)}$ to $\boldsymbol{L}^{(k)}$:

$$d_{k-1} = \beta_{k-1}/e_{k-1}, \quad e_k = \sqrt{\alpha_k - d_{k-1}^2}.$$

Here, $d_0 = 0$. Let $\boldsymbol{V}^{(k)} := \boldsymbol{B}\boldsymbol{Q}^{(k)}\boldsymbol{L}^{(k)-T}$. It is easy to see that the sequence $\boldsymbol{V}^{(k)}$ is nested, so that $\boldsymbol{V}^{(k)} = (\boldsymbol{v}^{(1)} \dots \boldsymbol{v}^{(k)})$. Moreover, we have the recurrence

$$\boldsymbol{v}^{(k)} = e_k^{-1}\left(\boldsymbol{B}\boldsymbol{q}^{(k)} - d_{k-1}\boldsymbol{v}^{(k-1)}\right),$$

which depends on the last recent $\boldsymbol{v}^{(k-1)}$ only. Finally,

$$\tilde{\boldsymbol{z}}^{(k)} = \mathrm{diag}^{-1}(\boldsymbol{V}^{(k)}\boldsymbol{V}^{(k)T}) = \boldsymbol{z}^{(k-1)} + \boldsymbol{v}^{(k)} \circ \boldsymbol{v}^{(k)}.$$

Since $z_i^{(k)} = z_i^{(k-1)} + (v_i^{(k)})^2$, the estimates $\tilde{\boldsymbol{z}}^{(k)}$ are nondecreasing in each component, and converge towards $\tilde{\boldsymbol{z}}$ from below. Recall from Section 4.5 that this monotonicity property has important implications in practice. In our implementation, we also estimate $\log|\boldsymbol{A}|$, required in the computation of $g^*(\boldsymbol{z})$, by $\log|\boldsymbol{T}^{(k)}|$ (the recurrence for $\boldsymbol{L}^{(k)}$ leads to a recurrence for these estimates), although this is not required in order to run our algorithms.

B.1 Lanczos Implementations

A word of warning: the Lanczos algorithm is a powerful method to approximate spectral information of very large structured matrices, but the simple three-point recurrence it is based on, hides much of the complications in practice. Without a number of additional mechanisms, significantly more difficult than the recurrence, an implementation on a finite-precision computer fails almost surely. Orthogonality is rapidly lost among the Lanczos vectors, which leads to the method getting stuck in a subspace. Ironically, this degradation is a byproduct of the recurrence being so successful: it is caused by Ritz vectors (estimates of eigenvectors of $\boldsymbol{A}$ in Krylov subspaces $\mathcal{K}_k$) converging. The full story, one of the fascinating ones in numerical mathematics, is given in (Parlett and Scott, 1979). But even the three-term recurrence itself is easily done wrong, as pointed out by Paige. We use the variant approved in (Paige, 1976).

We ignore preconditioning for now. Let $\boldsymbol{A}$ be the system matrix, and $\boldsymbol{c}$ be a unit norm starting vector, corresponding to the right hand side in linear conjugate gradients. The algorithm is given in Algorithm 1.

Algorithm 1 Lanczos algorithm

Require: Operator $\boldsymbol{A}$. Initial $\boldsymbol{c}$, $\|\boldsymbol{c}\| = 1$
  $\boldsymbol{q}^{(1)} = \boldsymbol{c}$. $\boldsymbol{u}^{(1)} = \boldsymbol{A}\boldsymbol{q}^{(1)}$
  for $k = 1, 2, \dots$ **do**
    $\alpha_k = \boldsymbol{q}^{(k)T}\boldsymbol{u}^{(k)}$
    Update estimates
    $\boldsymbol{r}^{(k)} = \boldsymbol{u}^{(k)} - \alpha_k\boldsymbol{q}^{(k)}$
    Re-orthogonalize $\boldsymbol{r}^{(k)}$
    $\beta_k = \|\boldsymbol{r}^{(k)}\|$. Stop if too small
    $\boldsymbol{q}^{(k+1)} = \boldsymbol{r}^{(k)}/\beta_k$
    $\boldsymbol{u}^{(k+1)} = \boldsymbol{A}\boldsymbol{q}^{(k+1)} - \beta_k\boldsymbol{q}^{(k)}$
  end for

From a practical viewpoint, the most important point in Algorithm 1 is the re-orthogonalization step: $\boldsymbol{r}^{(k)}$ (to become the new $\boldsymbol{q}^{(k+1)}$) has to be orthogonalized (or deflated) against *all* previous $\boldsymbol{q}^{(j)}$, $j \le k$. For a moderate number of iterations, this can be done naively, say by Gram-Schmidt, but we incur a cost of $O(n\,k^2)$ for this strategy, which for large k dominates the running time of the whole method. More advanced techniques have been proposed, where $\boldsymbol{r}^{(k)}$ is orthogonalized only against previously converged Ritz vectors (Parlett and Scott, 1979). However, they require more than the full matrix $\boldsymbol{Q}^{(k)}$ to be stored in memory, and eigendecompositions of some of the $\boldsymbol{T}^{(k)}$ have to be done. In most Lanczos applications, the spectrum of $\boldsymbol{A}$ decays geometrically, and queries of interest depend on few leading eigenvectors only. Generic codes such as `ARPACK` (see Section 4.5; *Matlab* `eigs` calls this code) seem to be tailored towards such spectral behaviour. If the system matrices $\boldsymbol{A}$ in a variational inference application are of this sort, these standard codes can be used. Unfortunately, as discussed in Section 4.5,

the spectral behaviour of $\boldsymbol{A}$ in natural image reconstruction problems is different, and standard codes may be slow or even fail. Our strategy in such cases is described at the end of this section.

Both LCG and Lanczos can typically be improved by *preconditioning*. The idea is to design some invertible $\boldsymbol{C}$, so that systems with $\boldsymbol{C}$ and $\boldsymbol{C}^T$ can be solved very rapidly (say, in $O(n)$), and at the same time, $\tilde{\boldsymbol{A}} = \boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{C}^{-T}$ has better properties as system matrix than $\boldsymbol{A}$ itself. For example, even the simple choice $\boldsymbol{C} = (\mathrm{diag}\,\boldsymbol{A})^{1/2}$ can lead to $\tilde{\boldsymbol{A}}$ being better conditioned than $\boldsymbol{A}$. If $\boldsymbol{A}$ is sparse, $\boldsymbol{C}$ can be computed by an incomplete Cholesky decomposition with small amount of fill-in. The Lanczos algorithm of Algorithm 1 can be run with a preconditioner $\boldsymbol{C}$, by simply replacing $\boldsymbol{A}$ by $\tilde{\boldsymbol{A}}$ and $\boldsymbol{c}$ by $\boldsymbol{C}^{-1}\boldsymbol{c}$ everywhere. The system matrices $\boldsymbol{A}$ in the application of our interest here are sufficiently well conditioned, and we do not use preconditioning at present. As noted in (Schneider and Willsky, 2001), preconditioning could in principle be used to improve spectral properties of $\tilde{\boldsymbol{A}}$ in comparison to $\boldsymbol{A}$. For special $\boldsymbol{A}$ coming from structured graphical models, such strategies have been proposed (see Section 4.5). However, in the image reconstruction applications of main interest here, such structure is not present. Research in preconditioners for these setups is an important topic for future work.

Lanczos with Lazy Selective Orthogonalization

The dilemma of linear spectral decay of $\boldsymbol{A}$ in applications of our interest here has been discussed in Section 4.5. The comments at the end of that section show that the resulting inaccuracy of Lanczos estimates do not invalidate reconstruction estimates or Bayesian design scores. However, other applications, or similar applications on different data, may rely more strongly on accurate posterior variance estimates.

We do not know of numerical mathematics work discussing this problem. Standard codes such as `ARPACK` just fail in these cases, or are very slow. Once more, the problem is not that leading eigenvectors of $\boldsymbol{A}$ are not found, but rather (ironically) that they converge rapidly. Time and memory requirements of modern Lanczos codes scale badly with the number of converged Ritz vectors. But the variance estimates we require depend significantly on a large part of the eigenspectrum of $\boldsymbol{A}$: they will be accurate only once many Ritz vectors have converged. Without additional structural knowledge about $\boldsymbol{A}$, there is probably little hope for principled improvement. But even for the specific structure of main interest here (image reconstruction from non-local Fourier measurements and local finite difference gradient potentials), which is heavily used in medical image reconstruction, we do not know of any prior analysis of LCG or Lanczos that would be helpful here.

Our present solution is to use Lanczos with complete (naive) re-orthogonalization, whenever the total running time is not dominating by the deflations. This variant is easiest to code and requires the least amount of memory. We use Gram-Schmidt deflation.[25] Our implementation also contains a modification of selective re-orthogonalization (Parlett and Scott, 1979), which is faster than the naive approach, but not drastically so (in the image reconstruction applications). In the language of Parlett and Scott (1979), a direct implementation of their scheme pauses in almost every iteration, and the running time is (perversely!) dominated by the eigendecompositions of $\boldsymbol{T}^{(k)}$ (required to test for converged Ritz vectors), and the re-computation of these vectors. Moreover, the strategies to avoid frequent pauses given there basically do not work in our case. A closer look reveals that their bounds are approximate themselves, relying on a favourable spectral decay of $\boldsymbol{A}$ as well. Our most important modification is that at a pause, previously converged Ritz vectors are assigned to present ones, without having to re-compute them. This assignment strategy is based on the assumption that a converged Ritz vector will appear at all subsequent pauses just as well. We still have to recompute the eigendecomposition of $\boldsymbol{T}^{(k)}$ at every pause.[26] Second, we do not use the heuristics of (Parlett and Scott, 1979), since they are not reliable in our case, but rather allow for pauses only at every fourth (or so) iteration.

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.

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

E. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23:969–985, 2006.

[25] Some experiments showed that Householder orthogonalization, advocated in (Golub and Van Loan, 1996), is not more accurate in our case, but runs significantly slower, presumably because fast BLAS primitives can be exploited to a lesser degree.

[26] Even though $\boldsymbol{T}^{(k)}$ is tridiagonal, this needs an iterative computation. We have a very good initial guess about this decomposition from the last recent pause, which could be used (Gu and Eisenstat, 1994).

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.

P. Chaudhuri and P. Mykland. Nonlinear experiments: Optimal design and inference based on likelihood. *Journal of the American Statistical Association*, 88(422):538–546, 1993.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Roy. Stat. Soc. B*, 39:1–38, 1977.

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.

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.

S. Gerwinn, J. Macke, M. Seeger, and M. Bethge. Bayesian inference for spiking neuron models with a sparsity prior. In Platt et al. (2008).

Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In Leen et al. (2001), pages 507–513.

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.

G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

M. Gu and S. Eisenstat. A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM. J. Matrix Anal.*, 15(4):1266–1276, 1994.

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

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

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.

R. E. Kaas and A. E. Raftery. Bayes factors and model uncertainty. *Journal of the American Statistical Association*, 90:773–795, 1995.

T. Leen, T. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems 13*, 2001. MIT Press.

D. Malioutov, J. Johnson, and A. Willsky. Low-rank variance estimation in large-scale GMRF models. In *Proceedings of ICASSP*, 2006a.

D. Malioutov, J. Johnson, and A. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7:2031–2064, 2006b.

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

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

C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *IMA Journal of Applied Mathematics*, 18(3):341–349, 1976.

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.

B. Parlett and D. Scott. The Lanczos algorithm with selective orthogonalization. *Mathematics of Computation*, 33 (145):217–238, 1979.

J. Platt, D. Koller, Y. Singer, and S. Roweis, editors. *Advances in Neural Information Processing Systems 20*, 2008. MIT Press.

J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli. Image denoising using Gaussian scale mixtures in the wavelet domain. *IEEE Transactions on Image Processing*, 12:1338–1351, 2003.

R. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

M. Schneider and A. Willsky. Krylov subspace estimation. *SIAM Journal on Scientific Computing*, 22(5):1840–1864, 2001.

M. Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

M. Seeger and H. Nickisch. Compressed sensing and Bayesian experimental design. In A. McCallum, S. Roweis, and R. Silva, editors, *International Conference on Machine Learning 25*. Omni Press, 2008.

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*, 2007.

M. Seeger, H. Nickisch, R. Pohmann, and B. Schölkopf. Bayesian experimental design of magnetic resonance imaging sequences. To appear at Neural Information Processing Systems 21, 2008.

E. Simoncelli. Modeling the joint statistics of images in the Wavelet domain. In *Proceedings 44th SPIE*, pages 188–195, 1999.

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.

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. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In C. Bishop and B. Frey, editors, *Workshop on Artificial Intelligence and Statistics 9*, 2003. Electronic Proceedings (ISBN 0-9727358-0-1).

M. Tipping and N. Lawrence. Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *NeuroComputing*, 69:123–141, 2005.

M. Wainwright, E. Sudderth, and A. Willsky. Tree-based modeling and estimation of Gaussian processes on graphs with cycles. In Leen et al. (2001), pages 661–667.

Y. Weiss, H. Chang, and W. Freeman. Learning compressed sensing. Snowbird Learning Workshop, Allerton, CA, 2007.

M. West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 1987.

D. Wipf and S. Nagarajan. A new view of automatic relevance determination. In Platt et al. (2008).

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.

A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.