Convex Variational Bayesian Inference for Large Scale Generalized Linear Models

Hannes Nickisch HN@TUEBINGEN.MPG.DE
Empirical Inference Department, Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Matthias W. Seeger MSEEGER@MMCI.UNI-SAARLAND.DE
Department of Computer Science, Saarland University, Saarbrücken, Germany

Abstract

We show how variational Bayesian inference can be implemented for very large generalized linear models. Our relaxation is proven to be a convex problem for any log-concave model. We provide a generic double loop algorithm for solving this relaxation on models with arbitrary super-Gaussian potentials. By iteratively decoupling the criterion, most of the work can be done by solving large linear systems, rendering our algorithm orders of magnitude faster than previously proposed solvers for the same problem. We evaluate our method on problems of Bayesian active learning for large binary classification models, and show how to address settings with many candidates and sequential inclusion steps.

1. Introduction

Modern machine learning applications require methods for inferring good decisions from incomplete data, in settings with many unknown variables. Bayesian inference is a powerful technique towards this end, but most current Bayesian algorithms cannot reliably operate in large scale domains, where convex point estimation techniques are routinely used.

We present a Bayesian inference approximation which can be used on a scale previously achieved by point estimation techniques only. We show that our variational relaxation constitutes a convex optimization problem, whenever the search for the posterior mode is convex. We propose a novel double loop algorithm for inference in large generalized linear models, reaching scalability by decoupling the criterion and reducing all efforts to standard techniques from numerical linear algebra. We generalize the algorithm from (Seeger et al., 2009) to arbitrary super-Gaussian sites and provide technology for a fully generic implementation. We show how our method applies to binary classification Bayesian active learning, and extend the framework of (Seeger et al., 2009) so to cope with thousands of sequential inclusions.

The structure of the paper is as follows: Our novel inference algorithm is presented in Section 2, along with theoretical results. Its application to binary classification active learning is given in Section 3, and relations to other work are discussed in Section 4. We present large scale experiments in Section 5. Much of the technical details are given in the Appendix.

2. Scalable Convex Inference

Our method applies to generalized linear models over continuous latent variables $\mathbf{u} \in \mathbb{R}^n$. More specifically, models may have Gaussian sites $N(\mathbf{y}|\mathbf{Xu}, \sigma^2\mathbf{I})$ and non-Gaussian factors $t_i(s_i)$, $\mathbf{s} = \mathbf{Bu}$.
For example, applied to magnetic resonance imaging (Seeger et al., 2009), $\mathbf{u}$ is the unknown MR image, $\mathbf{y} = \mathbf{Xu} + \boldsymbol{\varepsilon} \in \mathbb{C}^n$ are scanner measurements, and $t_i(s_i)$ form a sparsity prior on multiscale image gradients $\mathbf{s} = \mathbf{Bu} \in \mathbb{R}^q$.
In binary classification, $\mathbf{u}$ are classifier weights, $\mathbf{B}$ collects feature vectors $\mathbf{b}_i$, and $t_i(s_i)$ are Bernoulli likelihoods. For a Gaussian weight prior, $\mathbf{X} = \mathbf{I}$, $\mathbf{y} = \mathbf{0}$. A sparsity prior on $\mathbf{u}$ leads to $\mathbf{X} = \mathbf{0}$, $\mathbf{y} = \mathbf{0}$, appending $\mathbf{I}$ to $\mathbf{B}$, and adding sparsity sites to the $t_i(s_i)$.

The posterior of the model has the form

$$P(\mathbf{u}|\mathcal{D}) = P(\mathcal{D})^{-1}\mathcal{N}(\mathbf{y}|\mathbf{Xu}, \sigma^2\mathbf{I}) \prod_{i=1}^{q} t_i(s_i), \quad (1)$$

where $P(\mathcal{D})$ is a normalization constant. We re-

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).

fer to a site $t_i(s_i)$ as (strongly) *super-Gaussian* if $\log[t_i(s_i)e^{-\sigma^{-2}\beta_i s_i}]$ is even for some $\beta_i \in \mathbb{R}$, and strictly convex and decreasing as a function of s_i^2 (Palmer et al., 2006). For example, the sparsity promoting *Laplace* sites

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

are super-Gaussian with $\beta_i = 0$. For binary classification, *Bernoulli* (or logistic) sites are frequently used:

$$t_i(s_i) = \left(1 + e^{-c_i(\tau_i/\sigma)s_i}\right)^{-1}, \quad c_i \in \{\pm 1\}, \tag{3}$$

which are super-Gaussian as well (Jaakkola, 1997, Sect. 3.B) with $\beta_i = c_i\tau_i\sigma/2$ (see Appendix). Laplace and Bernoulli sites are *log-concave*: $s_i \mapsto \log t_i(s_i)$ is concave. This property, which is shared by many other potentials commonly used, will have important consequences. In the context of applications of interest here, Bayesian inference amounts to approximating the mean and covariance matrix of $P(\mathbf{u}|\mathcal{D})$. All of n, q, m can be hundreds of thousands. Our algorithms operate on this scale by exploiting structure in the model matrices $\mathbf{X}$, $\mathbf{B}$, so that *matrix-vector multiplications* (MVMs) can be computed rapidly, as is the case for sparse matrices, wavelet, or Fourier transforms.

Super-Gaussian sites can be lower-bounded by scaled Gaussians of any width. Let $x_i := \sigma^{-2}s_i^2$ and $g_i(x_i) := \log t_i(s_i) - \sigma^{-2}\beta_i s_i$ where g_i is convex and decreasing, so that $g_i(x_i) = \max_{\gamma_i>0}[-x_i/(2\gamma_i) - h_i(\gamma_i)/2]$ by Legendre-Fenchel duality (Palmer et al., 2006), and $t_i(s_i) = \max_{\gamma_i>0} e^{\sigma^{-2}(\beta_i s_i - s_i^2/(2\gamma_i)) - h_i(\gamma_i)/2}$. Here, h_i scales the height of the Gaussian lower bound such that it just touches the site $t_i(s_i)$ from below. To simplify notation, we will write $g_i(s_i) = g_i(x_i)$ in the following (g_i as dependent variable, rather than function). Variational Bayesian inference centers on the log-partition function $\log P(\mathcal{D})$ (1), the cumulant generating function of the posterior. The inference relaxation we employ here (Girolami, 2001; Palmer et al., 2006; Jaakkola, 1997) consists of lower-bounding $\log P(\mathcal{D})$, by plugging in the bounds for each site:

$$P(\mathcal{D}) \geq e^{-h(\boldsymbol{\gamma})/2} \int \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{u}, \sigma^2\mathbf{I})Q_0(\mathbf{u})\,d\mathbf{u}, \tag{4}$$

$Q_0(\mathbf{u}) := e^{\sigma^{-2}(\boldsymbol{\beta}^\top\mathbf{s} - \mathbf{s}^\top\boldsymbol{\Gamma}^{-1}\mathbf{s}/2)}$, $\boldsymbol{\Gamma} = \operatorname{diag}\boldsymbol{\gamma}$ and $h(\boldsymbol{\gamma}) = \sum_i h_i(\gamma_i)$. This is a tractable Gaussian integral, and previous algorithms maximize it one γ_i at a time (Girolami, 2001). If both n and q are very large, such coordinate ascent methods cannot be used anymore. Our novel contributions to this problem are twofold: First, we prove that this variational lower bound relaxation is a *convex* optimization problem iff all sites $t_i(s_i)$ are log-concave. Second, we provide a *scalable* algorithm to solve it orders of magnitude faster than previous algorithms. Our methods allow for the application of Bayesian techniques to problems, where previously only point estimation had been feasible, as will be demonstrated in our experiments.

2.1. Convex Inference Relaxation

The log-partition function lower bound implies a Gaussian approximate posterior $Q(\mathbf{u}|\mathcal{D})$ with covariance

$$\operatorname{Cov}_Q[\mathbf{u}|\mathcal{D}] = \sigma^2\mathbf{A}^{-1}, \quad \mathbf{A} := \mathbf{X}^\top\mathbf{X} + \mathbf{B}^\top\boldsymbol{\Gamma}^{-1}\mathbf{B}.$$

We start with an equality for Gaussians:

$$\int \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{u}, \sigma^2\mathbf{I})Q_0(\mathbf{u})\,d\mathbf{u} = |2\pi\sigma^2\mathbf{A}^{-1}|^{1/2}\max_{\mathbf{u}}\mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{u}, \sigma^2\mathbf{I})Q_0(\mathbf{u}),$$

so that the lower bound can be written as $P(\mathcal{D}) \geq C_1 e^{-\phi(\boldsymbol{\gamma})/2}$, with

$$\phi(\boldsymbol{\gamma}) := \log|\mathbf{A}| + h(\boldsymbol{\gamma}) + \min_{\mathbf{u}} R(\mathbf{u}, \boldsymbol{\gamma}),$$
$$R := \sigma^{-2}\left(\|\mathbf{y} - \mathbf{X}\mathbf{u}\|^2 + \mathbf{s}^\top\boldsymbol{\Gamma}^{-1}\mathbf{s} - 2\boldsymbol{\beta}^\top\mathbf{s}\right),$$

to be minimized w.r.t. $\boldsymbol{\gamma} \succ \mathbf{0}$. $R(\mathbf{u}, \boldsymbol{\gamma})$ is jointly convex, since $(s_i, \gamma_i) \mapsto s_i^2/\gamma_i$ is jointly convex for $\gamma_i > 0$. We will establish that $\boldsymbol{\gamma} \mapsto \log|\mathbf{A}|$ is convex as well, and that $\boldsymbol{\gamma} \mapsto h(\boldsymbol{\gamma})$ is convex iff all $t_i(s_i)$ are log-concave. We will show how to solve $\min_{\boldsymbol{\gamma}} \phi$ efficiently.

Theorem 1 *Let $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\mathbf{B} \in \mathbb{R}^{q\times n}$, and $f_i(\gamma_i)$ continuously differentiable functions into $\mathbb{R}_+$, so that $\log f_i(\gamma_i)$ are convex. Then, $\boldsymbol{\gamma} \mapsto \log|\mathbf{X}^\top\mathbf{X} + \mathbf{B}^\top(\operatorname{diag} f(\boldsymbol{\gamma}))\mathbf{B}|$ is convex. Especially, $\boldsymbol{\gamma} \mapsto \log|\mathbf{A}|$ is convex.*

A proof is given in the Appendix. The conditions are fulfilled for $f_i(\gamma_i) = \gamma_i^{-\alpha_i}$, $\alpha_i > 0$ ($\alpha_i = 1$ for the case $\log|\mathbf{A}|$), but also for $f_i(\gamma_i) = e^{\gamma_i}$, generalizing the convexity of the logsumexp function to matrix values.

Theorem 2 *Consider a model of the form (1), with strongly super-Gaussian sites, and let $g_i(x_i) = g_i(s_i) = \log t_i(s_i) - \sigma^{-2}\beta_i s_i$, $x_i = \sigma^{-2}s_i^2$. If $s_i \mapsto g_i(s_i)$ is concave and twice continuously differentiable for $s_i > 0$, then $h_i(\gamma_i) = -\min_{x_i\geq 0}(x_i/\gamma_i + 2g_i(x_i))$ is convex. On the other hand, if $g_i''(s_i) > 0$ for some $s_i > 0$, then $h_i(\gamma_i)$ is not convex at some $\gamma_i > 0$.*

A proof is given in the Appendix. We conclude from these theorems that the minimization of $\phi(\boldsymbol{\gamma})$ is a convex problem if all sites are strongly super-Gaussian and log-concave. On the other hand, for a model of the form (1), if some site $t_i(s_i)$ fails to be log-concave, then $h_i(\gamma_i)$ is not convex, and we can easily construct some $\mathbf{X}$, $\mathbf{y}$, $\mathbf{B}$ such that $\phi(\boldsymbol{\gamma})$ is not convex.

Our result settles a longstanding problem in approximate inference: If the posterior mode of a super-Gaussian model can be found via a convex problem, then a frequently used approximation (Girolami, 2001; Palmer et al., 2006; Jaakkola, 1997) is convex as well.

2.2. Scalable Double Loop Algorithm

Convexity of $\min_{\boldsymbol{\gamma}} \phi$ does not necessarily imply tractability on a large scale. For example, high resolution image reconstruction problems of the kind addressed in (Seeger et al., 2009) could not be sensibly approached by previous coordinate descent algorithms (Girolami, 2001): each γ_i update requires the corresponding marginal $Q(s_i|\mathcal{D})$, for which an $n \times n$ linear system has to be solved, and each of the q sites has to be visited at least once. If q and n are very large, such algorithms are intractable even for highly structured matrices. Standard joint optimization code is problematic as well, since even a single gradient $\nabla_{\boldsymbol{\gamma}}\phi$ is expensive. The algorithm we propose here, circumvents these problems by decoupling the expensive parts of ϕ in a nested double loop fashion, related to concave-convex (Yuille & Rangarajan, 2003) or difference of convex ideas, as used in expectation maximization. While double loop algorithms have been proposed for non-convex approximate inference, we show that they can also be used to drastically speed up solving *convex* inference problems.

We proceed as in (Seeger et al., 2009). While $\boldsymbol{\gamma} \mapsto \log|\mathbf{A}|$ is convex (Theorem 1), concavity of $\boldsymbol{\gamma}^{-1} \mapsto \log|\mathbf{A}|$ is well known. Using conjugate duality once more, $\log|\mathbf{A}| = \min_{\mathbf{z}\succ\mathbf{0}} \mathbf{z}^\top(\boldsymbol{\gamma}^{-1}) - g^*(\mathbf{z})$. We have

$$\phi \le \min_{\mathbf{u}} \phi_{\mathbf{z}}, \quad \phi_{\mathbf{z}} := \mathbf{z}^\top(\boldsymbol{\gamma}^{-1}) + h(\boldsymbol{\gamma}) + R(\mathbf{u},\boldsymbol{\gamma}) - g^*(\mathbf{z}).$$

The upper bound $\phi_{\mathbf{z}}$ is jointly convex in $(\mathbf{u},\boldsymbol{\gamma})$, and compared to ϕ, the difficult coupling term $\log|\mathbf{A}|$ is replaced by the decoupled convex term $\mathbf{z}^\top(\boldsymbol{\gamma}^{-1})$. Our algorithm proceeds in *inner loop minimizations* of $\phi_{\mathbf{z}}$ for fixed $\mathbf{z}$, and *outer loop updates*, where $\mathbf{z}$ and $g^*(\mathbf{z})$ are reestimated. For the inner loop, we note that

$$\min_{\boldsymbol{\gamma}} \phi_{\mathbf{z}} = 2\sigma^{-2}\left(\frac{1}{2}\|\mathbf{y}-\mathbf{X}\mathbf{u}\|^2 + \sum_{i=1}^{q} h_i^*(s_i)\right),$$

$h_i^*(s_i) = \frac{1}{2}\sigma^2 \min_{\gamma_i}((z_i + \sigma^{-2}s_i^2)/\gamma_i + h_i(\gamma_i)) - \beta_i s_i$. The problem $\min_{\mathbf{u}}(\min_{\boldsymbol{\gamma}} \phi_{\mathbf{z}})$ is of standard penalized quadratic form, and can be solved very efficiently by the *iteratively reweighted least squares* (IRLS) algorithm, which typically converges in few Newton steps, each of which requires the solution of one linear system $(\mathbf{X}^\top\mathbf{X}+\mathbf{B}^\top(\operatorname{diag}\mathbf{e})\mathbf{B})\mathbf{d} = \mathbf{r}$, where $\mathbf{r}$, $\mathbf{e}$ are simple functions of $\mathbf{u}$. Given useful structure in $\mathbf{X}$, $\mathbf{B}$ (such as sparsity), this optimization is scalable to very large sizes, the systems are solved by (preconditioned) linear conjugate gradients (LCG). Upon inner loop convergence, the minimizer $\mathbf{u}_*$ is the mean of $Q(\mathbf{u}|\mathcal{D})$.

Once an inner loop converges, $\mathbf{z}$ has to be refitted. The minimizer is $\mathbf{z}_* = \nabla_{\boldsymbol{\gamma}} \log|\mathbf{A}| = \operatorname{diag}^{-1}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^\top) = \sigma^{-2}(\operatorname{Var}_Q[s_i|\mathcal{D}])$, the marginal *variances* of $Q(\mathbf{s}|\mathcal{D})$, and $g^*(\mathbf{z}_*) = \mathbf{z}_*^\top(\boldsymbol{\gamma}^{-1}) - \log|\mathbf{A}|$. While the *means* of a large linear-Gaussian model $Q(\mathbf{u}|\mathcal{D})$ can be estimated by a single linear system, the *variances* are much harder to obtain. In fact, we do not know of a general bulk variance estimator which would be as accurate as LCG, but not vastly more expensive. To understand the rationale behind our algorithm, note that the computation of $\nabla_{\boldsymbol{\gamma}}\phi$ is as difficult as the estimation of $\mathbf{z}$. Our algorithm requires these expensive steps only few times (usually 4 or 5 outer loop iterations are sufficient), since they are kept out of the inner loop, where most of the progress is made. In contrast, most standard gradient-based optimizers require many evaluations of $\nabla_{\boldsymbol{\gamma}}\phi$ to converge. As discussed below, our decomposition also means that the variances can be estimated rather poorly, while still obtaining a practically useful algorithm. Finally, we note that our double loop algorithm is guaranteed to converge, under the assumption that the $\mathbf{z}$ are fitted exactly (Seeger et al., 2009; Wipf & Nagarajan, 2008).

For the inner loop optimization, we require $h_i^*(s_i)$ and its first and second derivatives. For Laplace sites, $h_i(\gamma_i) = \tau_i^2\gamma_i$, and $h_i^*(s_i) = \sigma^2\tau_i(z_i + s_i^2/\sigma^2)^{1/2}$. However, for Bernoulli sites, we are not aware of an analytic expression for $h_i(\gamma_i)$, let alone $h_i^*(s_i)$. Since h_i and h_i^* are defined by scalar convex minimizations, all terms can be computed implicitly whenever required. We show in the Appendix how to implement our algorithm generically, given $g_i(x_i)$ and its derivatives only. Even with many implicitly defined h_i^*, this can be done efficiently. Moreover, these computations can be parallelized straightforwardly.

2.3. Covariance Estimation by Lanczos

Outer loop updates require the computation of $\mathbf{z} = \operatorname{diag}^{-1}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^\top)$, or all variances $\operatorname{Var}_Q[s_i|\mathcal{D}]$, which we estimate by the Lanczos algorithm (Schneider & Willsky, 2001). In a nutshell, the precision matrix $\mathbf{A}$ is approximated by a low-rank representation $\mathbf{Q}\mathbf{T}\mathbf{Q}^\top$, $\mathbf{Q} \in \mathbb{R}^{n\times k}$ orthonormal, $\mathbf{T}$ tridiagonal, and $k \ll n$, where the extremal eigenvectors of $\mathbf{A}$ are optimally approximated by $\mathbf{Q}$. We then approximate expressions linear in $\mathbf{A}^{-1}$ by plugging in $\mathbf{Q}\mathbf{T}^{-1}\mathbf{Q}^\top$ instead, for example $\hat{\mathbf{z}}_k := \operatorname{diag}^{-1}(\mathbf{B}\mathbf{Q}\mathbf{T}^{-1}\mathbf{Q}^\top\mathbf{B}^\top) \approx \mathbf{z}$. In fact, $\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k-1} + \mathbf{v}_k^2$, with $\mathbf{v}_k = (\mathbf{B}\mathbf{q}_k - d_{k-1}\mathbf{v}_{k-1})/e_k$ (where d_k, e_k form the bidiagonal Cholesky factor of $\mathbf{T}$), so that $\hat{\mathbf{z}}_k$ increases monotonically towards $\mathbf{z}$ in each component. In this usage, the Lanczos algorithm can be thought of as solving many linear system in parallel, with the same $\mathbf{A}$ but different right hand sides.

This approach is not straightforward. Lanczos codes are complicated and scale superlinearly in the number of iterations k. The algorithm can be run with moderate k only, since at least $\mathcal{O}(n\,k)$ memory is required, and many components in $\hat{\mathbf{z}}_k$ are significantly underestimated. This inaccuracy may be unavoidable: we are not aware of a general bulk variances estimator improving on Lanczos, and variances are required to drive any algorithm for $\min_{\boldsymbol{\gamma}} \phi$. Importantly, these errors in $\hat{\mathbf{z}}_k \to \mathbf{z}$ seem to not much affect our algorithm in practice. Inaccurate variances mean that $\phi_{\mathbf{z}}$ is not exactly tangent to ϕ at the current $\boldsymbol{\gamma}$ after an outer loop update. However, its (inner loop) minimization is accurate, since mean computations are required only. In short, our algorithm seems to be rather robust in practice towards significant errors in posterior variance estimation. Given the apparent intractability of this computation, this is a critical feature of our decoupling approach. Some intuition about this robustness will be given in a longer paper. Compared to what is done in other tractable inference approximations, where many dependencies are ruled out up front without even inspecting the data (factorization assumptions in structured mean field), our approximation is fully data-dependent, with the extremal covariance eigenvectors being homed in by Lanczos, just as is done in PCA.

2.4. Properties of the Algorithm

The scalability of our algorithm comes from a number of appropriate *reductions* illustrated in Figure 1. On the first level, the complicated inference problem is relaxed to a convex program. The optimization problem is decoupled in the double loop algorithm: iterations reduce to the estimation of means and variances in a linear-Gaussian model, which is done by standard algorithms of numerical mathematics (LCG and Lanczos), routinely employed for very large systems. These naturally reduce to matrix-vector multiplications (MVMs). On a higher level, we fit a sequence of Gaussian models to the true posterior. The computational complexity of the algorithm is measured in number of MVMs needed, and can be related to MAP estimation and a naive approach to minimizing $\phi(\boldsymbol{\gamma})$.

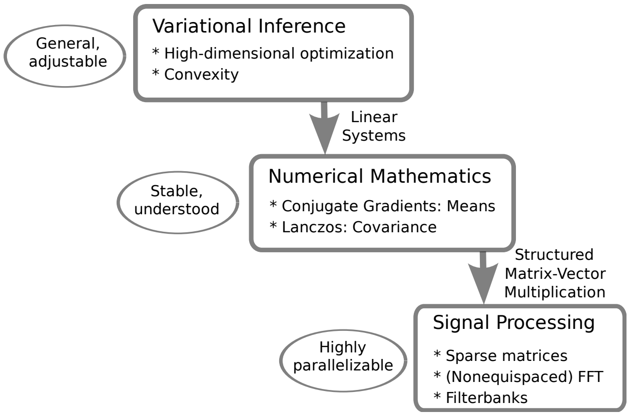

Figure 1. Reductions in Variational Inference

Recall that n is the number of latent variables, m the number of Gaussian, and q the number of non-Gaussian sites. Denote by k the number of Lanczos iterations in outer loop updates, by N_{CG} the number of LCG iterations to solve a system with $\mathbf{A}$, and by N_{Newt} the number of Newton steps for IRLS.

algorithm	# MVMs	storage
full Newton for MAP	$N_{\text{Newt}} \cdot N_{\text{CG}}$	$\mathcal{O}(m+n+q)$
one coordinate ascent step ϕ	$q \cdot N_{\text{CG}}$	$\mathcal{O}(m+n+q)$
one exact $\nabla_{\boldsymbol{\gamma}}\phi$	$q \cdot N_{\text{CG}}$	$\mathcal{O}(m+n+q)$
one approx $\nabla_{\boldsymbol{\gamma}}\phi$	$k + N_{\text{CG}}$	$\mathcal{O}(k \cdot n+q)$
one $(\mathbf{u}, \boldsymbol{\gamma})$ for inner loop	$N_{\text{Newt}} \cdot N_{\text{CG}}$	$\mathcal{O}(n+q)$
one $\mathbf{z}$ for outer loop	k	$\mathcal{O}(k \cdot n+q)$

The cost of an MVM for sparse matrices is typically linear in the number of nonzeros. Empirically, $N_{\text{Newt}} \approx 10$ for our inner loops, and we never run more than 5 outer loop iterations, typically 1 or 2 only. Lanczos codes come with additional costs to keep $\mathbf{Q}$ orthonormal, up to $\mathcal{O}(n \cdot k^2)$. The table shows that a naive minimization of $\phi(\boldsymbol{\gamma})$ is not scalable, since we have to solve $\mathcal{O}(q)$ $n \times n$ linear systems for a single gradient step. While MAP estimation is faster in practice, its scaling differs from our algorithm's only by a moderate constant factor.

3. Bayesian Active Learning

Proper Bayesian active learning builds on the posterior distribution, therefore inference is needed. More specifically, the evaluation of typical scores, such as the information gain, requires (approximate) inference. Within our method, these computations are naturally dealt with as part of the (convex) relaxation, without requiring artificial additional Laplace approximations. Our efficient double loop algorithm lets us address large scale Bayesian optimal decision problems. A special case was used in (Seeger et al.,

2009) in order to optimize measurements for image reconstruction. Their framework can only be run feasibly for a moderate number of sequential design extensions. We present extensions in order to deal with binary classification active learning, with weights $\mathbf{u}$, features given by the rows of $\mathbf{B}$, retaining scalability even with very many sequential inclusions. The likelihood sites $t_i(s_i)$ are Bernoulli (3), and the prior on $\mathbf{u}$ is a spherical Gaussian with variance σ^2 ($\mathbf{X} = \mathbf{I}$, $\mathbf{y} = \mathbf{0}$). If $n \geq q$, a sparsity prior on the weights may be more appropriate, and we employ such a variant in our experiments as well, using n Laplace sites (2) in addition to the likelihood sites. Technically, this is done by appending $\mathbf{I}$ to $\mathbf{B}$ and removing the Gaussian factor. For simplicity of notation, the Gaussian prior case is discussed in the remainder of this section.

Our active learning algorithm starts with a posterior approximation based on randomly drawn instances. In the subsequent design phase, we sequentially include blocks of K data points. If the task requires a large number of sequential inclusions, tractability is retained by choosing K large enough. We are using a candidate list of potential rows for $\mathbf{B}$. Each iteration consists of an initial Lanczos run to estimate marginal posterior moments, $K \geq 1$ inclusions (appending K new rows to $\mathbf{B}$), and a re-optimization of all site parameters $\boldsymbol{\gamma}$. Within a block, the marginals $Q(s_i) = \mathcal{N}(\mu_i, \sigma^2 \rho_i)$, $i \in J$ containing all model and candidate sites, are kept valid at all times. Note that $\boldsymbol{\mu}_J = \mathbf{B}_J \mathbf{u}_*$ (since $\mathbf{u}_* = \mathrm{E}_Q[\mathbf{u}|\mathcal{D}]$), and that $\mathbf{B}$ is a part of $\mathbf{B}_J$. For larger K, our method runs faster, since the variational parameters $\boldsymbol{\gamma}$ are updated less frequently, while for smaller K, the more frequent refits to the non-Gaussian posterior may result in better sequential decisions.

Each inclusion within a block consists of scoring all remaining candidates, picking the winner, and updating the marginals $\boldsymbol{\mu}_J$, $\boldsymbol{\rho}_J$. Let $\mathbf{b}_i$ be a potential new row of $\mathbf{B}$, and $s_i = \mathbf{b}_i^\top \mathbf{u}$. In our experiments, we use several design scores, based on the current marginal $Q(s_i)$. The information gain score (Seeger, 2008, Sect. 4.1) is $S_{\mathrm{IG}}(\mathbf{b}_i) := \sum_{c_i = \pm 1} Q(c_i) D[Q'(s_i; c_i) \,\|\, Q(s_i)]$, where $Q'(\cdot; c_i)$ is the new approximation to $\propto Q(s_i) t_i(s_i; c_i)$. The classifier uncertainty score is simpler: $S_{\mathrm{CU}}(\mathbf{b}_i) := -|Q(c_i = +1) - 1/2|$, preferring candidates with predictive probability $Q(c_i = +1)$ closest to $\frac{1}{2}$. We compute $Q(c_i = +1) = \mathrm{E}_Q[t_i(s_i; c_i = +1)]$ by quadrature.

Once the winner is determined (say, i), its label c_i is queried and its site included, using the novel γ_i (detailed in the Appendix). The marginals are updated as suggested in (Seeger & Nickisch, 2008): $\boldsymbol{\rho}'_J = \boldsymbol{\rho}_J - (\rho_i + \gamma_i)^{-1} \mathbf{w}^2$, $\boldsymbol{\mu}'_J = \boldsymbol{\mu}_J + ((\beta_i - \mu_i/\gamma_i)/\kappa_i)\mathbf{w}$, where $\kappa_i = 1 + \rho_i/\gamma_i$ and $\mathbf{w} = \mathbf{B}_J \mathbf{d}$, $\mathbf{d} = \mathbf{A}^{-1} \boldsymbol{\beta}_i$ (one linear system). We use the solution to recompute ρ_i, μ_i, solve again for γ_i, and plug these back into $\boldsymbol{\mu}_J$, $\boldsymbol{\rho}_J$. This corrects for Lanczos inaccuracies (especially ρ_i is underestimated by Lanczos). Moreover, $\mathbf{u}'_* = \mathbf{u}_* + ((\beta_i - \mu_i/\gamma_i)/\kappa_i)\mathbf{d}$, and $\log|\mathbf{A}'| = \log|\mathbf{A}| + \log \kappa_i$.

At the end of a block, we re-run our variational algorithm in order to update all site parameters jointly (within a block, only γ_i for novel model sites are updated). In practice, a single outer loop iteration suffices for these runs. Importantly, the first outer loop update comes for free, since the model marginals (part of $\boldsymbol{\mu}_J$, $\boldsymbol{\rho}_J$), $\mathbf{u}_*$, and $\log|\mathbf{A}|$ have been kept valid. Therefore, only a single Lanczos run per block is required. Finally, since variances are underestimated by Lanczos, it may happen that components in $\boldsymbol{\rho}_J$ become negative within a block. Such components are simply removed, and if they correspond to model sites, their marginals are recomputed by solving linear systems at the end of the block.

While there is some complexity to our scheme, note that the principal computational primitives are always the same: solving linear systems with $\mathbf{A}$ (or simple variants thereof), and variance estimation by Lanczos based on $\mathbf{A}$. This is discussed in more detail in Section 2.4.

4. Related Work

Active learning can be done using a large variety of criteria. For an empirical review and collection of heuristics see (Schein & Ungar, 2007). We use Bayesian active learning, meaning that the scores for inclusion decisions are computed based on the posterior distribution, which seems the approach favoured by decision theory in general. Given that, we can employ a host of different scores, and the particular ones used in our experiments here could certainly be improved upon by heuristic experience with the task. However, the general consensus is that to compute Bayesian scores, such as the information gain, accurately over many different candidates, Bayesian inference (in particular pertaining to the posterior covariance) has to be approximated well.

Our lower bound variational inference approximation has been used many times before (Girolami, 2001; Jaakkola & Jordan, 1997; Palmer et al., 2006). Our scalable double loop algorithm is novel, as is the characterization of the relaxation as a convex problem. Either have been given for the special case of Laplace sites in (Seeger et al., 2009). We extend these results to generalized linear models with arbitrary super-Gaussian sites, and give a generic implementation which can be run as is with any kind of super-Gaussian sites, which are either log-concave, or for which $h_i(\gamma_i)$ can be worked out analytically. Moreover, their method can be applied to problems with many inclusions only with the additional technology presented here.

5. Experiments

We use three standard data sets for binary classification[1], outlined in the table below. The feature vectors are sparse, and a MVM with the matrix $\mathbf{B}$ costs $\mathcal{O}(\#\mathrm{nz})$.

Dataset	q	q_+/q_-	n	# nonzeros
`a9a`	$32,561$	0.32	123	$451,592$
`real-sim`	$72,201$	0.44	$20,958$	$3,709,083$
`rcv1`	$677,399$	1.10	$42,736$	$49,556,258$

We randomly select 16, 36 and 50 thousand instances for training; the rest is kept for testing. Hyperparameters τ_{bern}, σ^2, and τ_{lapl} were determined on the full datasets. Results are given in Figure 2. We ran sparse logistic regression (with Laplace prior) on `a9a` only, results for the larger sets will be included in the final version of this paper. As expected, our algorithm runs longer in this case, and is less tolerant w.r.t. larger block sizes K: the Laplace prior site parameters have to be updated in response to new cases, in order to do their job properly. Although sparse classification improves on the Gaussian prior case beyond about 2800 cases, active learning works better with a Gaussian prior for fewer inclusions. This may be due to the case that the Lanczos variance estimation is exact for $q < k$, and in general more accurate in the Gaussian prior case. Over all sets, we see clear improvements of active learning with the classifier uncertainty score S_{CU} over random sampling of data cases. Somewhat surprisingly, the information gain score does much less well in the binary classification case. Since sequential experimental design with this score performs well in other cases (Seeger et al., 2009), a closer analysis of the respective merits of different design scores w.r.t. different statistical tasks is an important point for future research.

6. Discussion

We have shown that a frequently used variational relaxation to Bayesian inference in super-Gaussian models is convex if and only if the posterior is log-concave. Our double loop inference algorithm decouples the criterion and runs orders of magnitude faster than previous coordinate descent methods. Computational efforts are reduced to fast algorithms of numerical mathematics and exploiting fast MVMs due to the structured matrices $\mathbf{X}$ and $\mathbf{B}$. Our generic implementation, can be run with any configuration of super-Gaussian, log-concave sites. We have pointed out the crucial role of the Lanczos algorithm for bulk variance estimation in large Gaussian models, an essential step for variational approximate inference methods. Our framework can be used for a multitude of Bayesian decision and experimental design problems, and is most useful for tasks where the choice of good decisions depends on posterior covariance information.

Appendix

Proof of Theorem 1

Let $\mathbf{A} = \mathbf{X}^\top \mathbf{X} + \mathbf{B}^\top f(\mathbf{\Gamma})\mathbf{B}$ for now, $\psi_1 = \log|\mathbf{A}|$. $d\psi_1 = \operatorname{tr}\mathbf{SD}$, $\mathbf{S} = \mathbf{BA}^{-1}\mathbf{B}^\top$ and $\mathbf{D} = f'(\mathbf{\Gamma})(d\mathbf{\Gamma})$. Then, $d^2\psi_1 = -\operatorname{tr}\mathbf{SDSD} + \operatorname{tr}\mathbf{S}f''(\mathbf{\Gamma})(d\mathbf{\Gamma})^2 = \operatorname{tr}\mathbf{DSD}(p(\mathbf{\Gamma}) - \mathbf{S})$, where $p_i(\gamma_i) = f_i''(\gamma_i)/(f_i'(\gamma_i))^2$. Since $\mathbf{S} \succeq \mathbf{0}$ (positive semi-definite): $\mathbf{S} = \mathbf{VV}^\top$ with some $\mathbf{V}$, and $d^2\psi_1 = \operatorname{tr}(\mathbf{DV})^\top(p(\mathbf{\Gamma}) - \mathbf{S})\mathbf{DV}$. If we show that $p(\mathbf{\Gamma}) - \mathbf{S} \succeq \mathbf{0}$, then for $\boldsymbol{\gamma}^t = \boldsymbol{\gamma} + t(\Delta\boldsymbol{\gamma})$: $\psi_1''(0) = \operatorname{tr}\mathbf{M}^\top(p(\mathbf{\Gamma}) - \mathbf{S})\mathbf{M} \geq 0$ for all small $\Delta\boldsymbol{\gamma}$, where $\mathbf{M} = f'(\mathbf{\Gamma})(\Delta\mathbf{\Gamma})\mathbf{V}$, so that ψ_1 is convex.

We show that $f(\mathbf{\Gamma})^{-1} - \mathbf{S} \succeq \mathbf{0}$, employing the identity $\mathbf{r}^\top\mathbf{M}^{-1}\mathbf{r} = \max_{\mathbf{x}} 2\mathbf{r}^\top\mathbf{x} - \mathbf{x}^\top\mathbf{Mx}$, $\mathbf{M} \succ \mathbf{0}$ (positive definite). Now, $\mathbf{r}^\top\mathbf{BA}^{-1}\mathbf{B}^\top\mathbf{r} = \max_{\mathbf{x}} 2\mathbf{r}^\top\mathbf{Bx} - \mathbf{x}^\top\mathbf{Ax} \leq \max_{\mathbf{y}=\mathbf{Bx}} 2\mathbf{r}^\top\mathbf{y} - \mathbf{y}^\top f(\mathbf{\Gamma})\mathbf{y}$, since $\mathbf{x}^\top\mathbf{X}^\top\mathbf{Xx} = \|\mathbf{Xx}\|^2 \geq 0$. Taking the maximum over all $\mathbf{y} \in \mathbb{R}^q$, we see that $\mathbf{r}^\top\mathbf{Sr} \leq \max_{\mathbf{y}} 2\mathbf{r}^\top\mathbf{y} - \mathbf{y}^\top f(\mathbf{\Gamma})\mathbf{y} = \mathbf{r}^\top f(\mathbf{\Gamma})^{-1}\mathbf{r}$, so that $f(\mathbf{\Gamma})^{-1} - \mathbf{S} \succeq \mathbf{0}$. We are done if $p(\mathbf{\Gamma}) - f(\mathbf{\Gamma})^{-1} \succeq \mathbf{0}$, which is equivalent to $f_i(\gamma_i)f_i''(\gamma_i) \geq (f_i'(\gamma_i))^2$ for all i, which in turn is equivalent to all $\log f_i(\gamma_i)$ being convex

Proof of Theorem 2

We focus on a single site i and drop its index. Since $b \neq 0$ is dealt with separately, assume that $t(s)$ is even itself. Since positive scaling does not influence convexity, assume that $\sigma^2 = 1$ in this section only, so that $g(s) = g(x) = \log t(s)$, $x = s^2$. By conjugate duality: $h(\gamma) = \max_{x\geq 0} f(x,\gamma)$, $f := -x/\gamma - 2g(x)$. We only consider $s \geq 0$. Let $x_* := \operatorname{argmax}_{x\geq 0} f(x,\gamma)$ (unique, since $g(x)$ is strictly convex). If $\gamma_0 := \sup\{\gamma \,|\, f(s,\gamma) \leq -2g(0)\ \forall s\}$ ($\gamma_0 = 0$ for an empty set), then $x_* = 0$, $h(\gamma) = -2g(0)$ for $0 < \gamma \leq \gamma_0$, and for $\gamma > \gamma_0$, h is strictly increasing and $x_* > 0$. Since $g(x)$ is convex and decreasing, conjugate duality implies that for

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

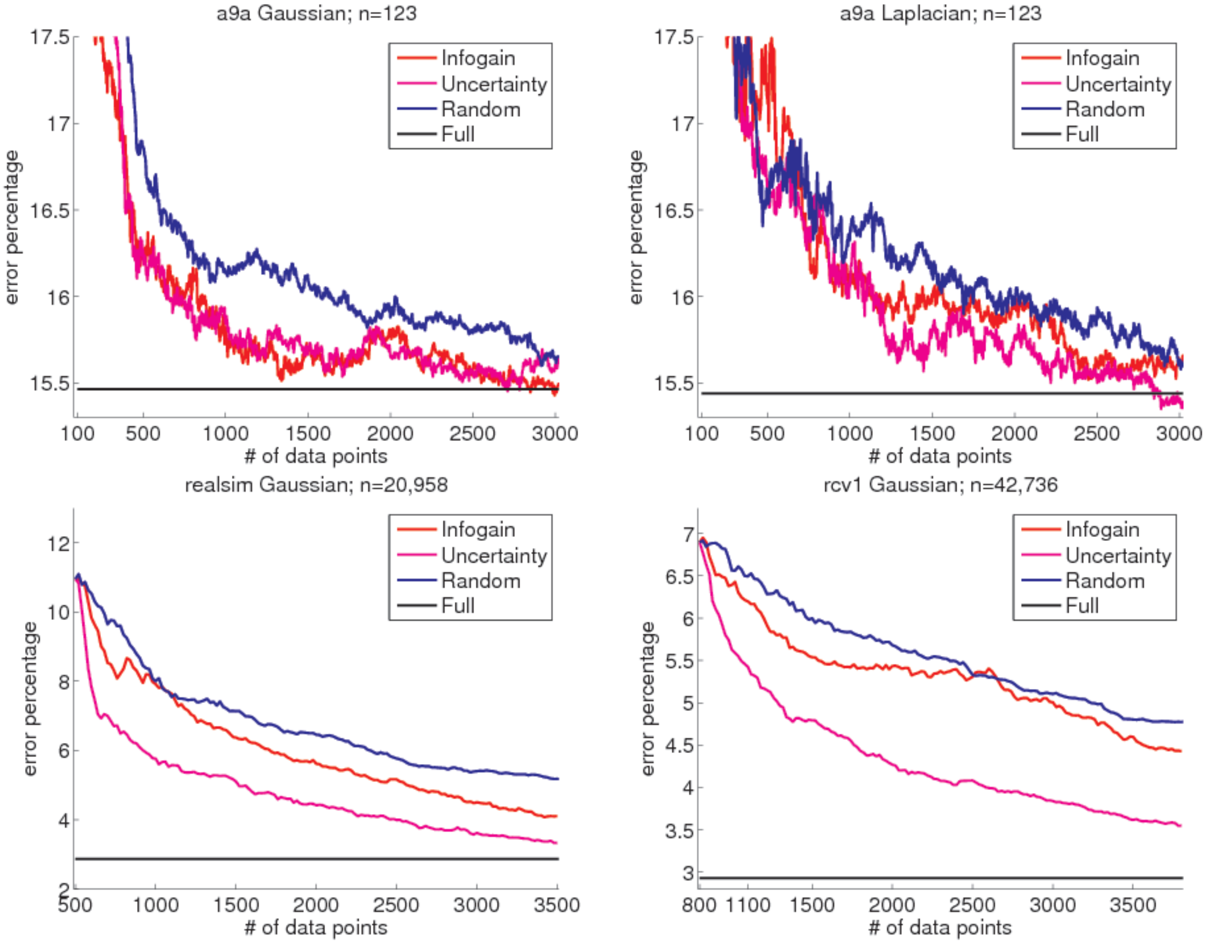

Figure 2. Classification errors for different design scores (information gain, classifier uncertainty), vs. random sampling (results on full training set also shown). Design started after 100, 100, 500, 800 randomly drawn initial cases respectively, all remaining training cases were candidates. Prior variance $\sigma^2 = 1$ in all cases, $\tau_{\text{bern}} = 1, 1, 3, 3$ respectively. $k = 80, 80, 750, 750$ Lanczos vectors for outer loop updates/candidate scoring. For `a9a`, we used design blocks of size $K = 3$, and $K = 20$ for the others.

every $x > 0$, there exists some γ: $x_*(\gamma) = x$. It suffices to show that h is convex at all $\gamma > \gamma_0$, where $x_* = s_*^2 > 0$.

We use the notation $f_s = \partial f/(\partial s)$, functions are evaluated at (x_*, γ) if nothing else is said. Now, $f_s = -2s_*/\gamma - 2g_s(s_*) = 0$, so that $g_s(s_*) = -s_*/\gamma$. Next, $x \mapsto g(x)$ is twice continuously differentiable, and $x_* = s_*^2$ at γ. Therefore, $f_x = \partial f/(\partial x)$ is continuously differentiable, and $g_{x,x}(x) > 0$ by the strict convexity of $g(x)$. By the implicit function theorem, $x_*(\gamma)$ is continuously differentiable at γ, and since $h(\gamma) = f(x_*(\gamma), \gamma)$, $h'(\gamma)$ exists. Moreover, $0 = (d/d\gamma) f_x(x_*(\gamma), \gamma) = f_{x,\gamma} + f_{x,x} \cdot (dx_*)/(d\gamma)$, so that $(dx_*)/(d\gamma) = \gamma^{-2}/(2g_{x,x}(x_*)) > 0$: $x_*(\gamma)$ is increasing. From $f_s = 0$, we have that $h'(\gamma) = f_\gamma = s_*^2/\gamma^2 = (g_s(s_*))^2$, since $g_s(s_*) = -s_*/\gamma$. Now, $g_s(s)$ is nonincreasing by the concavity of $g(s)$, and $g_s(s_*) < 0$, so that $s_* \mapsto h'(\gamma)$ is nondecreasing. Since $s_*^2 = x_*$ is increasing in γ, so is s_*. Therefore, $\gamma \mapsto h'(\gamma)$ is nondecreasing, and $h(\gamma)$ is convex for $\gamma > \gamma_0$.

The concavity of $g(s)$ is necessary as well. Suppose that $g_{s,s}(\tilde{s}) > 0$ for some $\tilde{s} > 0$. Now, $g_s(s_*) < 0$ whenever $s_* > 0$, and there exists some $\tilde{\gamma} > 0$ so that $s_*(\tilde{\gamma}) = \tilde{s}$. But if $g_{s,s}(s_*) > 0$ at $\tilde{\gamma}$, then $s_* \mapsto h'(\gamma)$ is decreasing at $s_* = \tilde{s}$, and just as above $\gamma \mapsto h'(\gamma)$ is decreasing at $\tilde{\gamma}$, so that h is not convex at $\tilde{\gamma}$.

Implicit Computation of h_i and h_i^*

We focus on a single site i and drop its index. Our method requires the evaluation of $h^*(s) = \frac{1}{2}\sigma^2(\min_\gamma k(x,\gamma)) - bs$, $k := (z + x)/\gamma + h(\gamma)$, $x = s^2/\sigma^2$, as well as $(h^*)'(s)$ and $(h^*)''(s)$ (for the inner loop Newton steps). Let $\gamma_* = \operatorname{argmin} k(x,\gamma)$. Assuming for now that h and its derivatives are available, γ_* is found by univariate Newton minimization[2], where $\gamma^2 k_\gamma = -(z+x)+\gamma^2 h'(\gamma)$, $\gamma^3 k_{\gamma,\gamma} = 2(z+x)+\gamma^3 h''(\gamma)$. Now, $k_\gamma = 0$ (always evaluated at (x, γ_*)), so that $(h^*)'(s) = s/\gamma_* - b$. Moreover, $0 = (d/ds)k_\gamma = k_{s,\gamma} + k_{\gamma,\gamma} \cdot (d\gamma_*)/(ds)$, so that $(h^*)''(s) = \gamma_*^{-1}(1 - s\gamma_*^{-1}(d\gamma_*)/(ds)) = \gamma_*^{-1}(1 - 2x/(\gamma_*^3 k_{\gamma,\gamma}(x, \gamma_*)))$.

Next, by Fenchel-Legendre duality, $h(\gamma) = -\min_x l(x, \gamma)$, $l := x/\gamma + 2g(x)$, where $g(x)$ is strictly convex and decreasing. We need methods to evaluate $g(x)$ and its first and second derivatives ($g''(x) > 0$). The minimizer $x_* = x_*(\gamma)$ is found by convex minimization once more, started from the last recently found x_* for this site, where $\gamma l(x, \gamma)$ should be minimized instead of $l(x, \gamma)$. Note that $x_* = 0$ iff $\gamma \le \gamma_0 := -1/(2g'(0))$, which has to be checked up front. Given x_*, we have that $\gamma h(\gamma) = -x_* - 2\gamma g(x_*)$. Since $l_x = 0$ for $\gamma > \gamma_0$ (always evaluated at (x_*, γ)), we have that $\gamma^2 h'(\gamma) = -\gamma^2 l_\gamma = x_*$ (this holds even if $l_x > 0$ and $x_* = 0$). Moreover, if $x_* > 0$ (for $\gamma > \gamma_0$), then $(d/d\gamma) l_x(x_*, \gamma) = 0$, so that $(dx_*)/(d\gamma) = \gamma^{-2}/(2g''(x_*))$, and $\gamma^3 h''(\gamma) = (2\gamma g''(x_*))^{-1} - 2x_*$. If $x_* = 0$ and $l_x > 0$, then $x_*(\tilde{\gamma}) = 0$ for $\tilde{\gamma}$ close to γ, so that $h''(\gamma) = 0$. A critical case is $x_* = 0$ and $l_x = 0$, which happens for $\gamma = \gamma_0$: $h''(\gamma)$ does not exist at this point in general. This is not a problem for our code, since we employ a robust Newton/bisection search for γ_*. If $\gamma > \gamma_*$, but is very close, we approximate $x_*(\gamma)$ by $\xi_0(\log\gamma - \log\gamma_0)$ with $\xi_0 := -g'(0)/g''(0)$.

For Bernoulli sites (3), we have that $g(x) = -\log\cosh(v) - \log 2$, $v := (y\tau/2)x^{1/2} = (y\tau/2)\sigma^{-1}|s|$, so that $g'(x) = -C(\tanh v)/v$, $g''(x) = (C/2)x^{-1}((\tanh v)/v + \tanh^2 v - 1)$, $C = (y\tau/2)^2/2$. Care has to be taken when evaluating these for $x \approx 0$. Moreover, $\gamma_0 = 1/(2C)$ and $\xi_0 = 3/(2C)$.

Active Learning Scores

Recall Section 3. For the classifier uncertainty score S_{CU}, we require $Q(c_i = +1) = \mathrm{E}_Q[t(s_i; c_i)]$, which we approximate by Gaussian quadrature. The simple information gain score S_{IG} is computed as follows. If the candidate $\boldsymbol{\beta}_i$ is scored under the label assumption c_i, then the offset is $\beta_i = c_i \tau_i \sigma/2$. The lower bound to $P(\mathcal{D} \cup \{(\boldsymbol{\beta}_i, c_i)\})$ is

$$e^{-h_i(\gamma_i)/2} \mathrm{E}_Q \left[e^{\sigma^{-2}(\beta_i s_i - s_i^2/(2\gamma_i))} \right] \propto e^{-\phi_i(\gamma_i)/2}$$

up to a constant not depending on γ_i, and simple calculations give $\phi_i(\gamma_i) = h_i(\gamma_i) + \log\kappa_i - \sigma^{-2}(\mu_i + \rho_i\beta_i)^2/(\rho_i\kappa_i)$, $\kappa_i := 1 + \rho_i/\gamma_i$, where $Q(s_i) = \mathcal{N}(\mu_i, \sigma^2\rho_i)$. This convex function is minimized just as $\gamma \mapsto k(x, \gamma)$ in the previous subsection. If $Q'(s_i) \propto Q(s_i) e^{\sigma^{-2}(\beta_i s_i - \frac{1}{2}s_i^2/\gamma_i)}$ at the minimizer γ_i, then

$$D[Q' \,\|\, Q] = \frac{1}{2}\left(\log\kappa_i + \frac{\rho_i}{\kappa_i}\left(\frac{(\beta_i - \mu_i/\gamma_i)^2}{\sigma^2\kappa_i} - \gamma_i^{-1}\right)\right).$$

The score $S_{\mathrm{IG}}(\boldsymbol{\beta}_i)$ is obtained by averaging over $Q(c_i)$.

[2] We find a root of k_γ, by first constructing a bracket, then using a robust combination of Newton and bisection steps.

References

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

Jaakkola, T. (1997). *Variational methods for inference and estimation in graphical models*. Doctoral dissertation, Massachusetts Institute of Technology.

Jaakkola, T., & Jordan, M. (1997). Improving the mean field approximation via the use of mixture distributions. In M. I. Jordan (Ed.), *Learning in graphical models*. Kluwer.

Palmer, J., Wipf, D., Kreutz-Delgado, K., & Rao, B. (2006). Variational em algorithms for non-Gaussian latent variable models. *Advances in Neural Information Processing Systems 18* (pp. 1059–1066).

Schein, A. I., & Ungar, L. H. (2007). Active learning for logistic regression: An evaluation. *Machine Learning*, *68*, 235–265.

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

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

Seeger, M., & Nickisch, H. (2008). Compressed sensing and Bayesian experimental design. *International Conference on Machine Learning 25* (pp. 912–919).

Seeger, M., Nickisch, H., Pohmann, R., & Schölkopf, B. (2009). Bayesian experimental design of magnetic resonance imaging sequences. *Advances in Neural Information Processing Systems 21* (pp. 1441–1448).

Wipf, D., & Nagarajan, S. (2008). A new view of automatic relevance determination. *Advances in Neural Information Processing Systems 20* (pp. 1625–1632).

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