Journal of Machine Learning Research 14 (2013) 807-829 Submitted 3/12; Revised 10/12; Published 3/13

Quasi-Newton Methods: A New Direction

Philipp Hennig PHILIPP.HENNIG@TUEBINGEN.MPG.DE
Martin Kiefel MARTIN.KIEFEL@TUEBINGEN.MPG.DE
Department of Empirical Inference
Max Planck Institute for Intelligent Systems
Spemannstraße 38
Tübingen, Germany

Editor: Manfred Opper

Abstract

Four decades after their invention, quasi-Newton methods are still state of the art in unconstrained numerical optimization. Although not usually interpreted thus, these are learning algorithms that fit a local quadratic approximation to the objective function. We show that many, including the most popular, quasi-Newton methods can be interpreted as approximations of Bayesian linear regression under varying prior assumptions. This new notion elucidates some shortcomings of classical algorithms, and lights the way to a novel nonparametric quasi-Newton method, which is able to make more efficient use of available information at computational cost similar to its predecessors.

Keywords: optimization, numerical analysis, probability, Gaussian processes

1. Introduction

Quasi-Newton algorithms are arguably the most popular class of nonlinear numerical optimization methods, used widely in numerical applications not just in machine learning. Their defining property is that they iteratively build estimators B_i for the Hessian $B(x) = \nabla\nabla^\intercal f(x)$ of the objective function $f(x)$, from observations of f's gradient $\nabla f(x)$, at each iteration searching for a local minimum along a line search direction $-B_i^{-1}\nabla f(x)$, an estimate of the eponymous Newton-Raphson search direction. Some of the most widely known members of this family include Broyden's (1965) method, the SR1 formula (Davidon, 1959; Broyden, 1967), the DFP method (Davidon, 1959; Fletcher and Powell, 1963) and the BFGS method (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). Decades of continued research effort in this area make it impossible to give even a superficial overview over the available literature. The textbooks by Nocedal and Wright (1999) and Boyd and Vandenberghe (2004) are good modern starting points for readers interested in background. An insightful and extensive contemporary review was compiled by Dennis and Moré (1977). The ubiquity of optimization problems in machine learning has made these algorithms tools of the trade. But, perhaps because they predate machine learning itself, they have rarely been studied as learning algorithms in their own right. This paper offers a probabilistic analysis.

Throughout, let $f : \mathbb{R}^N \to \mathbb{R}$ be a sufficiently regular, not necessarily convex, function; $\nabla f : \mathbb{R}^N \to \mathbb{R}^N$ its gradient; $B : \mathbb{R}^N \to \mathbb{R}^{N\times N}$ its Hessian. We consider iterative algorithms moving from location $x_{\ell-1} \in \mathbb{R}^D$ to location x_ℓ. The algorithm performs consecutive *line searches* along one-dimensional subspaces $x_i(\alpha) = \alpha e_i + x_i^0$, with $\alpha \in \mathbb{R}_+$ and a unit length vector $e_i \in \mathbb{R}^N$ spanning the line search space starting at x_i^0. Evaluations at x_i evince the gradient $\nabla f(x_i)$ (and usually also $f(x_i)$,

though this will not feature in this paper). The goal is to find a candidate x^* for a local minimum: a root $\nabla f(x^*) = 0$ of the gradient.

The derivations of classical quasi-Newton algorithms proceed along the following line of argument: We require an update rule incorporating an observation $\nabla f(x_{i+1})$ into a current estimate B_i to get a new estimate B_{i+1}, subject to the following desiderata:

1. **Low Rank/Cost Updates** Optimization problems regularly have dimensionality above $N \sim 10^3$, even beyond $N \sim 10^6$. To keep computational costs tractable, the update to the estimator B_i for the Hessian should be of the form

$$B_i = B_{i-1} + uCv^\top \qquad \text{with } u, v \in \mathbb{R}^{N\times M}, C \in \mathbb{R}^{M\times M},$$

   with low rank M (usually $M = 1$ or 2), because, by the matrix inversion lemma, its inversion, and multiplication with the gradient has (worst-case) cost $\mathcal{O}(N^2 + NM + M^3)$.

2. **Consistency with Quadratic Model** If f is locally described well to second order, then

$$y_i \equiv \nabla f(x_i) - \nabla f(x_{i-1}) \approx B(x_i)s_i, \tag{1}$$

   with $s_i \equiv x_i - x_{i-1}$. Because this is the fundamental idea behind this family of algorithms, it is also known as *the quasi-Newton equation*.

3. **Symmetry** The Hessian of twice differentiable functions is symmetric; so the estimator should be symmetric, too.

4. **Positive Definiteness** *Convex* functions have positive definite Hessians everywhere. Over time, it has become common conviction that, even for non-convex problems, positive definiteness of the estimator is desirable.

1.1 Outline

This first half of this paper (Section 2) constructs a new conceptual interpretation of quasi-Newton methods. Adopting a probabilistic viewpoint, we interpret the two requirements classically used to derive this family of methods as log likelihood and log prior, both of a specific Gaussian form. Varying the prior covariance and choosing one of two possible likelihoods gives rise to the different members of the family of quasi-Newton methods. A surprising insight arising from this analysis is that the way symmetry and positive definiteness (desiderata 3 and 4 above) are ensured in existing quasi-Newton methods differs from the way one would naïvely choose from the probabilistic perspective. In fact, the posterior arising from the newly identified prior and likelihood assigns nonzero probability mass to non-symmetric (Section 2.1), and to indefinite matrices (Section 2.2). It is only the maximum of the posterior, the estimator used by quasi-Newton methods, that is both symmetric and positive definite. Interestingly, the "proper" probabilistic way to ensure these properties has much higher computational complexity (Sections 2.1 and 3.5).

The second half of the paper (Section 3) uses the insights gained in Section 2 to construct a novel nonparametric Bayesian quasi-Newton algorithm. This replaces the approximate form of desideratum 2 above with an exact, analytic expression. We show that the structural ideas developed in Section 2 extend from the classic parametric formulation to a Gaussian Process model keeping computational cost *linear* in the input dimensionality (it has cost $\mathcal{O}(NM + M^3)$). A further advantage

of the nonparametric formulation is that it allows the use of every gradient observation calculated during a line search instead of just the last one, something that is not easily achievable under the old parametric models.

1.2 Notation

The derivations in the following sections require a compact notation for joint Gaussian probability densities over the elements of matrices. This often requires re-arranging the elements of a matrix $A \in \mathbb{R}^{N\times M}$ into a vector in $\mathbb{R}^{NM}$, which we denote by $\overrightarrow{A}$. We will assume this vectorization operation stacks the rows of A into column vector row by row, not column by column (this choice is relevant, it has effects Equation (3) below). Instead of introducing a new scalar index for the elements of such vectors, it will be convenient to keep the original indices i, j of the matrix A, and interpret them as an index set (ij) of the vector.

Throughout, we make use of the following sum convention: Indices that appear more than once on one side of an equation are summed over, unless they also appear on the other side of the equation. We also extensively use the Kronecker product. Given $A \in \mathbb{R}^{I\times K}$ and $B \in \mathbb{R}^{J\times L}$, the Kronecker product $A \otimes B \in \mathbb{R}^{IJ\times KL}$ has elements

$$(A \otimes B)_{(ij)(k\ell)} = A_{ik}B_{j\ell}. \tag{2}$$

In this notation, using the sum convention defined above, the vectorization of the matrix product AXB can be re-written as

$$\overrightarrow{(AXB)}_{ij} = A_{ik}X_{k\ell}B_{\ell j} = A_{ik}B^{\top}_{j\ell}X_{k\ell} = \left[(A \otimes B^{\top})\overrightarrow{X}\right]_{ij}. \tag{3}$$

Some important properties of Kronecker products are

$$\begin{aligned}
&(A \otimes B)(C \otimes D) = AC \otimes BD, && (A \otimes B)^{-1} = A^{-1} \otimes B^{-1},\\
&\alpha(A \otimes B) = (\alpha A \otimes B) = (A \otimes \alpha B), && \mathrm{rk}(A \otimes B) = \mathrm{rk}A \cdot \mathrm{rk}B,\\
&\mathrm{tr}(A \otimes B) = \mathrm{tr}A \cdot \mathrm{tr}B, && \det(A \otimes B) = \det{}^{\mathrm{rk}(B)}A \cdot \det{}^{\mathrm{rk}(A)}B.
\end{aligned}$$

The identities in the left column directly follow from (2), the less straightforward identities on the right can be found in matrix algebra collections (e.g., Lütkepohl, 1996; Minka, 2000a).

2. Quasi-Newton Methods as Approximate Bayesian Regressors

Aiming for a probabilistic interpretation of quasi-Newton methods, we consider them as regularised maximum likelihood (that is, maximum a posteriori) estimation schemes. The quasi-Newton equation (1) is a likelihood for B. Using $s_i = x_i - x_{i-1}$, we can write it using Dirac's distribution as

$$p(y_i|B, s_i) = \delta(y_i - Bs_i) = \lim_{\beta\to 0} \mathcal{N}\left[y_i; \mathcal{S}_{\triangleright}^{\top}\overrightarrow{B}, (V_{i-1} \otimes \beta)\right], \tag{4}$$

with any arbitrary $N \times N$ matrix V_{i-1}, a scalar β, and the linear operator $\mathcal{S}_{\triangleright} = (I \otimes s_i)$ (the significance of the subscript $\triangleright$ will become clear later). Instead of enforcing this point mass likelihood (4), we could equivalently minimize its negative logarithm

$$-\log p(y_i|B, s_i) = \lim_{\beta\to 0} \tfrac{1}{\beta}(y_i - Bs_i)^{\top}V^{-1}(y_i - Bs_i) + \text{const.}$$

Since the N real numbers in y_i are not sufficient to identify the N^2 numbers in B, classic derivations (Dennis and Moré, 1977; Nocedal and Wright, 1999) choose the estimator minimizing a *regularized* loss,

$$B_i = \operatorname*{argmin}_{B\in\mathbb{R}^{N\times N}} \left\{ \lim_{\beta\to 0} \tfrac{1}{\beta}(y_i - Bs_i)^\top V^{-1}(y_i - Bs_i) + \|B - B_{i-1}\|_{F,V_{i-1}^{-1}} \right\},$$

using the weighted Frobenius norm $\|\cdot\|_{F,V_{i-1}^{-1}}$ from the current best estimate B_{i-1} from previous iterations. The weight in the Frobenius norm is encoded using a positive definite matrix, which we will suggestively call V_{i-1}^{-1} and identify with the V_{i-1} of Equation (4)

$$\begin{aligned} \|B - B_{i-1}\|_{F,V_{i-1}^{-1}} &\equiv \operatorname{tr}(V_{i-1}^{-1}(B - B_{i-1})^\top V_{i-1}^{-1}(B - B_{i-1})) \\ &= (\overrightarrow{B} - \overrightarrow{B}_{i-1})^\top (V_{i-1}^{-1} \otimes V_{i-1}^{-1})(\overrightarrow{B} - \overrightarrow{B}_{i-1}). \end{aligned} \tag{5}$$

The new estimate is the unique matrix B_i minimizing the regularizer subject to Equation (4). Inspecting Equation (5) we see that, up to additive constants, the Frobenius regularizer is the negative logarithm of a Gaussian prior

$$p(B) = \mathcal{N}\left[\overrightarrow{B}; \overrightarrow{B}_{i-1}, \Sigma_{i-1} \equiv (V_{i-1} \otimes V_{i-1})\right]. \tag{6}$$

Gaussian likelihoods are conjugate to Gaussian priors (the sum of quadratic forms is a quadratic form). So the posterior is Gaussian, too, even for the limit case of a Dirac likelihood. We perform the following derivations for finite β, then take the limit at the end. A first form for the posterior can be found by explicitly multiplying the two Gaussians and "completing the square" in the exponent of the product of Gaussians: Posterior covariance and mean are

$$\begin{aligned} \Sigma_\triangleright &= (\Sigma_{i-1}^{-1} + \mathcal{S}_\triangleright (V_{i-1}^{-1} \otimes \beta^{-1}) \mathcal{S}_\triangleright^\top)^{-1}, \\ B_\triangleright &= \Sigma_\triangleright (\mathcal{S}_\triangleright (V_{i-1}^{-1} \otimes \beta^{-1}) \overrightarrow{Y} + \Sigma_{i-1}^{-1} \overrightarrow{B}_{i-1}). \end{aligned}$$

The following observation is helpful in the search for a more compact form (e.g., Rasmussen and Williams, 2006, Equation 2.12). Because $\Sigma_\triangleright$ is invertible for any finite β,

$$\begin{aligned} \mathcal{S}_\triangleright (V_{i-1}^{-1} \otimes \beta^{-1})(\mathcal{S}_\triangleright^\top \Sigma_{i-1} \mathcal{S}_\triangleright + V_{i-1} \otimes \beta) &= \Sigma_\triangleright^{-1} \Sigma_{i-1} \mathcal{S}_\triangleright, \\ \Sigma_\triangleright \mathcal{S}_\triangleright (V_{i-1}^{-1} \otimes \beta^{-1})(\mathcal{S}_\triangleright^\top \Sigma_{i-1} \mathcal{S}_\triangleright + V_{i-1} \otimes \beta) &= \Sigma_{i-1} \mathcal{S}_\triangleright, \\ \Sigma_\triangleright \mathcal{S}_\triangleright (V_{i-1}^{-1} \otimes \beta^{-1}) &= \Sigma_{i-1} \mathcal{S}_\triangleright (\mathcal{S}_\triangleright^\top \Sigma_{i-1} \mathcal{S}_\triangleright + V_{i-1} \otimes \beta)^{-1}. \end{aligned}$$

The step from the first to the second line is multiplication from the left by $\Sigma_\triangleright^{-1}$, the one from the second to the third is multiplication from the right by $(\mathcal{S}_\triangleright^\top \Sigma_{i-1} \mathcal{S}_\triangleright + V_{i-1} \otimes \beta)^{-1}$. Using this result, we re-write the posterior mean, using the Matrix inversion lemma, as

$$\begin{aligned} \overrightarrow{B}_\triangleright &= \Sigma_\triangleright ((V_{i-1}^{-1} \otimes \beta^{-1}) \mathcal{S}_\triangleright \overrightarrow{Y} + \Sigma_{i-1}^{-1} \overrightarrow{B}_{i-1}) \\ &= \overrightarrow{B}_{i-1} + \Sigma_{i-1} \mathcal{S}_\triangleright (\mathcal{S}_\triangleright^\top \Sigma_{i-1} \mathcal{S}_\triangleright + V_{i-1} \otimes \beta)^{-1} \cdot (\overrightarrow{Y} - \mathcal{S}_\triangleright^\top \overrightarrow{B}_{i-1}). \end{aligned}$$

Now we plug in the explicit expressions for $\mathcal{S}_\triangleright$ and Σ_{i-1}. Note that $\Sigma_{i-1}\mathcal{S}_\triangleright = (V_{i-1} \otimes V_{i-1})(I \otimes s_i) = (V_{i-1} \otimes V_{i-1}s_i)$ and likewise $\mathcal{S}_\triangleright^\top \Sigma_{i-1}\mathcal{S}_\triangleright = (V_{i-1} \otimes s_i^\top V_{i-1}s_i)$. So the posterior has mean and covariance

$$\begin{aligned} B_i &= B_{i-1} + \lim_{\beta \to 0} \frac{(y_i - B_{i-1}s_i)s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i + \beta} && = B_{i-1} + \frac{(y_i - B_{i-1}s_i)s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i} \quad \text{and} \\ \Sigma_i &= V_{i-1} \otimes \left(V_{i-1} - \lim_{\beta \to 0} \frac{V_{i-1}s_i s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i + \beta} \right) && = V_{i-1} \otimes \left(V_{i-1} - \frac{V_{i-1}s_i s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i} \right) \\ &\equiv V_{i-1} \otimes V_i, \end{aligned} \tag{7}$$

respectively. The new mean is a rank-1 update of the old mean, and the rank of the new covariance Σ_i is one less than that of Σ_{i-1}. The posterior mean has maximum posterior probability (minimal regularized loss), and is thus our new point estimate. Choosing a unit variance prior $\Sigma_{i-1} = I \otimes I$ recovers one of the oldest quasi-Newton algorithms: *Broyden's method* (1965):

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1}s_i)s_i^\top}{s_i^\top s_i}.$$

Broyden's method does not satisfy the third requirement of Section 1: the updated estimate is, in general, not a symmetric matrix. A supposed remedy for this problem, and in fact the *only* rank-1 update rule that obeys Equation (4) (Dennis and Moré, 1977) is the *symmetric rank 1 (SR1)* method (Davidon, 1959; Broyden, 1967):

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1}s_i)(y_i - B_{i-1}s_i)^\top}{s_i^\top (y_i - B_{i-1}s_i)}.$$

The SR1 update rule has acquired a controversial reputation (e.g., Nocedal and Wright, 1999, §6.2): While some authors report good results using this method, others note that it is unstable and overly limited. Our Bayesian interpretation identifies the SR1 formula as Gaussian regression with a data-dependent prior variance involving V_{i-1} with

$$V_{i-1}s_i = (y_i - B_{i-1}s_i).$$

Given the explicitly Gaussian prior of Equation (6), there is no rank 1 update rule that gives a symmetric posterior. This blemish of rank-1 updates is also reflected in Equation (7): Uncertainty drops only in the "row", or "primal" subspace of the belief (the right hand side of the Kronecker product in the covariance). While this still means uncertainty goes toward 0 over time, it does so in an asymmetric way.

2.1 Symmetric Estimates, but no Symmetric Beliefs

Many classic quasi-Newton methods provide symmetric estimators for B. Is it possible to encode the Hessians symmetry directly in the probabilistic belief? The proper probabilistic way to do so is to include an additional factor

$$\delta(\Delta\overrightarrow{B} - \overrightarrow{0}) = \lim_{\tau \to 0} \mathcal{N}(\overrightarrow{0}, \Delta\overrightarrow{B}, \tau I) \tag{8}$$

using Δ, the *antisymmetry* operator—the linear map defined through

$$\Delta\overrightarrow{X} = \frac{1}{2}\overrightarrow{(X - X^\top)}.$$

Since this is a linear map, the resulting posterior is analytic, and Gaussian. But the rank of Δ is $1/2 \cdot N(N-1)$ (e.g., Lütkepohl, 1996, §4.3.1, Equations 12 & 20), so the corresponding update rule does not obey the first requirement of Section 1. So, while it is possible to encode symmetry, it is not practical. However, the structure of Equation (7) hints at another idea, which in fact turns out to give rise to the most popular quasi-Newton methods. We introduce a second, *dual* observation (dual, as in "dual vector space", not as in "primal-dual optimization"), using the operator $\mathcal{S}_\triangleleft = (s_i \otimes I)$, the dual of $\mathcal{S}_\triangleright$,

$$p(y_i^\top \,|\, B, s_i^\top) = \delta(y_i^\top - s_i^\top B) = \lim_{\gamma \to 0} \mathcal{N}\left[y_i^\top; \mathcal{S}_\triangleleft^\top \overrightarrow{B}, (\gamma \otimes V_i)\right]. \tag{9}$$

Note that the limit uses V_i, not V_{i-1} as in Equation (4). The posterior has mean

$$\begin{aligned}
\overrightarrow{B}_i &= \overrightarrow{B}_\triangleright + \Sigma_\triangleright \mathcal{S}_\triangleleft \left(K_\triangleleft + \gamma I \otimes V_\triangleright\right)^{-1} \left(\overrightarrow{y_i^\top} - \mathcal{S}_\triangleleft^\top \overrightarrow{B}_\triangleright\right) \\
&= \overrightarrow{B}_{i-1} + \left(I \otimes \frac{V_{i-1} s_i}{s_i^\top V_{i-1} s_i + \beta}\right) \overrightarrow{(y_i - B_{i-1} s_i)} \\
&\quad + (V_{i-1} s_i \otimes V_i)\left[(s_i^\top V_{i-1} s + \gamma) \otimes V_i\right]^{-1} \overrightarrow{\left[y_i^\top - s_i^\top \left(B_{i-1} + \frac{y_i - B_{i-1} s_i}{s_i^\top V_{i-1} s_i + \beta} s_i^\top V_{i-1}\right)\right]}.
\end{aligned}$$

The calculation for the posterior covariance can be reduced to a simple symmetry argument. Expanding the Kronecker products as before, we find that the posterior after both primal and dual observation is a Gaussian with mean and covariance

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1} s_i) s_i^\top V_{i-1}^\top}{s_i^\top V_{i-1} s_i} + \frac{V_{i-1} s_i (y_i - B_{i-1} s_i)^\top}{s_i^\top V_{i-1} s_i} - \frac{V_{i-1} s_i (s_i^\top (y_i - B_{i-1} s_i)) s_i^\top V_{i-1}}{(s_i^\top V_{i-1} s_i)^2}, \tag{10}$$

$$\Sigma_i = \left(V_{i-1} - \frac{V_{i-1} s_i s_i^\top V_{i-1}}{s_i^\top V_{i-1} s_i}\right) \otimes V_i = V_i \otimes V_i. \tag{11}$$

The posterior mean is clearly symmetric if B_{i-1} is symmetric (as V_{i-1} is symmetric by definition). Choosing the unit prior $\Sigma_{i-1} = I \otimes I$ once more, Equation (10) gives what is known as *Powell's* (1970) *symmetric Broyden (PSB)* update. Equation (10) has previously been known to be the most general form of a symmetric rank 2 update obeying the quasi-Newton equation (1) and minimizing a Frobenius regularizer (Dennis and Moré, 1977). This old result is a corollary of our derivations. But note that symmetry only extends to the mean, not the entire belief: In contrast to the posterior generated by Equation (8), samples from this posterior are, with probability 1, not symmetric. Of course, they can be projected into the space of symmetric matrices by applying the symmetrization operator Γ defined by

$$\Gamma \overrightarrow{X} = \frac{1}{2} \overrightarrow{\left(X + X^\top\right)} \qquad (\text{note that } I = \Gamma + \Delta; \Gamma\Delta = 0). \tag{12}$$

Since Γ is a symmetric linear operator, the projection of any Gaussian belief $\mathcal{N}(X; X_0, \Sigma)$ onto the space of symmetric matrices is itself a Gaussian $\mathcal{N}(\Gamma X; \Gamma X_0, \Gamma\Sigma\Gamma)$. But symmetrized samples from the posterior of Equations (10), (11) do not necessarily obey the quasi-Newton Equation (1). While Equation (9) does convey useful information, it is not equivalent to encoding symmetry. It is cheaper, but also weaker, than using the likelihood (8), which encodes the full information afforded by symmetry.

2.2 Positive Definiteness: Meaning or Decoration?

So quasi-Newton methods ensure symmetry in the maximum of the posterior, but not the posterior itself. What about desideratum 4 from Section 1, positive definiteness? Consider choosing $V_{i-1} = B$. The prior (6) then turns into the *non-Gaussian* form

$$p(B) \propto |B|^{-N^2/2} \cdot \exp\left[-\frac{1}{2}\left(N - 2\operatorname{tr}(B_{i-1}B^{-1}) + \operatorname{tr}(B_{i-1}B^{-1}B_{i-1}B^{-1})\right)\right]. \tag{13}$$

This is an intriguing prior. The last term in the exponential has the form of the natural Riemannian metric on the space of positive definite real matrices (Savage, 1982), and may also remind some readers of the Wishart distribution. But the second term in the exponential means this prior is broader than the Wishart. It is not well-defined for degenerate matrices, and it is not clear whether it is proper. It is thus surprising to discover that it engenders the two most popular quasi-Newton methods: If we use the quasi-Newton equation (1) a second time to replace $V_{i-1}s = y$, Equation (13) gives the *DFP* method (Davidon, 1959; Fletcher and Powell, 1963)

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1}s_i)y_i^\top}{s_i^\top y_i} + \frac{y_i(y_i - B_{i-1}s_i)^\top}{y_i^\top s_i} - \frac{y_i(s_i^\top(y_i - B_{i-1}s_i))y_i^\top}{(y_i s_i)^2}.$$

And, if we exchange in the entire preceding derivation $s \leftrightarrow y$, $B \leftrightarrow B^{-1}$, $B_{i-1} \leftrightarrow B_{i-1}^{-1}$, then we arrive at the *BFGS* method (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), which ranks among the most widely used algorithms in numerical optimization. Table 1 gives an overview over the relationships between quasi-Newton methods described so far. It also mentions methods by Greenstadt (1970) and McCormick (see Pearson, 1969) which contain the "missing links" in this table but have not been mentioned so far. These works are also briefly discussed by Dennis and Moré (1977), from where we take these citations.

DFP and BFGS owe much of their popularity to the fact that the updated $B_{i,\mathrm{DFP}}$ and $B_{i,\mathrm{BFGS}}^{-1}$ are guaranteed to be positive definite whenever $B_{i-1,\mathrm{DFP}}$ and $B_{i-1,\mathrm{BFGS}}^{-1}$ are positive definite, respectively, and additionally $y_i^\top s_i > 0$. How helpful is this property? It is relatively straightforward to extend a theorem by Dennis and Moré (1977) to find that, assuming B_{i-1} is positive definite, the posterior mean of Equation (10) is positive definite if, and only if,

$$\begin{aligned} & 0 < (y_i^\top B_{i-1}^{-1} V_{i-1} s_i)^2 + (y_i - B_{i-1}s_i)^\top B_{i-1}^{-1} y_i \cdot s_i^\top V_{i-1} B_{i-1}^{-1} V_{i-1} s_i \\ \Leftrightarrow \quad & 0 < s_i^\top V_{i-1}[B_{i-1}^{-1} y_i y_i^\top B_{i-1}^{-1} - y^\top B_{i-1}^{-1} y_i + s_i^\top y_i] V_{i-1} s_i. \end{aligned}$$

If the prior covariance is not to depend on the data, it is thus impossible to guarantee positive definiteness in this framework—BFGS and DFP circumvent this conceptual issue by choosing $V_{i-1} = B$, then applying Equation (1) a second time. But, even casting aside such philosophical reservations, our analysis also casts doubt upon the efficacy of the way in which DFP and BFGS achieve positive definiteness: Equation (13) does not exclude indefinite matrices; in fact it assigns positive density to every invertible matrix. For example, under a mean $B_{i-1} = I$, the indefinite matrix $B = \operatorname{diag}(1, -1)$ is assigned $p(B) \propto \exp(-2)$. DFP and BFGS achieve positive definiteness, not by including additional information, but by manipulating the prior such that the *MAP estimator* (not the belief) happens to be positive definite. These observations do not rule out any utility of guaranteeing positive definiteness in this way, and the prior (13) deserves closer study. But these results suggest there is less value in the positive definiteness guarantee of DFP and BFGS than previously thought.

likelihood	prior	inferring B	inferring B^{-1}
$y = Bs$	$V = I$	Broyden (1965)	
	$V = B$	Pearson (1969)	McCormick
	$Vs = (y - Bs)$	SR1 (Davidon, 1959)	
$y = Bs \wedge y^\top = Bs^\top$	$V = I$	PSB (Powell, 1970)	Greenstadt (1970)
	$V = B$	DFP	BFGS

Table 1: Overview over probabilistic interpretations of various quasi-Newton methods, based on the combination of prior and likelihood. The "McCormick" entry refers to a note in Pearson (1969), see also Dennis and Moré (1977). The SR1 method is identical for inference on either B or its inverse. The abbreviation DFP stands for Davidon (1959), Fletcher and Powell (1963), while BFGS stands for Broyden (1970), Fletcher (1970), Goldfarb (1970) and Shanno (1970).

2.3 Rank M Updates

The classical quasi-Newton algorithms update the mean of the belief at every step in a rank 2 operation, then, implicitly, reset their uncertainty in the next step, thereby discarding information acquired earlier. Albeit inelegant from a Bayesian point of view, this scheme is still a good idea given other aspects of the framework: Since the quasi-Newton likelihood models the objective function as a quadratic, with constant Hessian everywhere, strict Bayesian inference from this prior would simply average over the Hessian everywhere, which is obviously not a good model. But it is instructive to consider the effect of encoding more than just the most recent observation. It is straightforward to extend Equation (4) to observations (Y, S) from several line searches:

$$Y_{nm} = \nabla_n f(x_{i_m}) - \nabla_n f(x_{i_m - 1}), \qquad S_{nm} = x_{i_m, n} - x_{i_m - 1, n}.$$

Given a prior $p(B) = \mathcal{N}(B; B_0, V_0)$, the Gaussian posterior then has mean and covariance

$$\begin{aligned} B_i &= B_0 + (Y - B_0 S)(S^\top V_0 S)^{-1} S^\top V_0 + V_0 S (S^\top V_0 S)^{-1} (Y - B_0 S)^\top \\ &\quad - V S (S^\top V_0 S)^{-1} (S^\top (Y - B_0 S)) (S^\top V S)^{-1} S^\top V_0, \\ \Sigma_i &= \left(V_0 - V_0 S (S^\top V_0 S)^{-1} S^\top V_0\right) \otimes \left(V_0 - V_0 S (S^\top V_0 S)^{-1} S^\top V_0\right). \end{aligned} \tag{14}$$

Here, the absence of information about the symmetry of the Hessian becomes even more obvious: No matter the prior covariance V_0, because of the term $S^\top Y$ in the second line of Equation (14), the posterior mean is not in general symmetric, *unless* $Y = BS$, (e.g., if the objective function is in fact a quadratic). See Section 4, particularly Figure 3, for a simple experiment with this parametric algorithm.

2.4 Summary

The preceding section showed that quasi-Newton algorithms, including the state-of-the-art BFGS and DFP algorithms, can be interpreted as approximate Bayesian regression from the primal and dual likelihood of Equations (4) and (9) under varying priors, in the following sense: At each quasi-Newton step, fix a Gaussian prior ad hoc, update the mean, then "forget" the covariance update.

Two particularly interesting observations concern the way in which the desiderata of symmetry and positive definiteness of the MAP estimator are achieved in these algorithms. Symmetry is encoded via dual observations, which is a useful but imperfect shortcut. Similarly, positive definiteness is just guaranteed for the mode, not the entire support of the posterior distribution. There may well be a non-obvious value to the "scale-free" prior of Equation (13) (see also Nocedal and Wright, 1999, Equations 6.11–6.13), but our analysis raises doubt on whether the proven good performance of BFGS and DFP is actually down to positive definiteness, or to a different effect involving the broader non-Gaussian prior (13).

3. A Nonparametric Bayesian Quasi-Newton Method

Section 2 used the probabilistic perspective to gain novel insight into classical methods. It showed that quasi-Newton methods can be interpreted as Gaussian regressors using algebraic structure to weaken prior knowledge, in exchange for lower computational cost. In this second part of the paper we depart from the traditional framework to construct a nonparametric, Bayesian quasi-Newton method, de novo. To motivate this effort, notice some other deficiencies of DFP/BFGS not directly connected to computational cost: Equation (4) assumes that the function is (locally) a quadratic. Old observations collected "far" from the current location (in the sense that a second order expansion is a poor approximation) may thus be useless or even harmful. The fact that the function is not quadratic should be part of the model. On an only slightly related point, individual line searches typically involve several evaluations of the objective function f and its gradient; but the algorithms only make use of one of those (the last one). This is clearly wasteful, but even the exact Bayesian parametric algorithm of Section 2.3 has this problem, because the matrix S of several observations along one line search has rank 1, so the inverse of $S^\top V_0 S$ is not defined. The following section will address all these issues, using the framework of nonparametric Gaussian process regression to model the objective function more closely.

3.1 A Nonparametric Prior

Defining a prior for the function $B : \mathbb{R}^N \to \mathbb{R}^{N\times N}$, we choose a set of N^2 correlated Gaussian processes. The mean function is assumed to be an arbitrary integrable function $B_0(x)$ (in our implementation we use a constant function, but the analytic derivations do not need to be so restrictive). The core idea is to assume that the covariance between the element B_{ij} at location[1] $x_\triangledown$ and the entry $B_{k\ell}$ at location $x_\triangle$ is

$$\operatorname{cov}\left(B_{ij}(x_\triangledown), B_{k\ell}(x_\triangle)\right) = k_{ik}(x_\triangledown^\top, x_\triangle^\top) k_{j\ell}(x_\triangledown, x_\triangle) = \left(k \otimes k\right)_{(ij)(k\ell)}(x_\triangledown, x_\triangle)$$

with an $N \times N$ matrix of kernels, k. To give a more concrete intuition: In our implementation we use one joint squared exponential kernel for all elements. I.e.

$$k_{ij}(x_\triangledown, x_\triangle) = V_{ij} \exp\left(-\frac{1}{2}(x_\triangledown - x_\triangle)^\top \Lambda^{-1}(x_\triangledown, x_\triangle)\right) \tag{15}$$

1. We use the notation $x_\triangle$ and $x_\triangledown$ (read "x up" and "x down") to denote two separate, arbitrary elements of the input space. The combinations x_* and x^* or x and x', or x_1 and x_2 are more widely used in the literature. But since this document is heavy on indices, we prefer this notation as it prevents confusion over sub- and superscripts and named indices.

with a positive definite matrix V and length scales Λ. This means

$$\operatorname{cov}(B_{ij}(x_\triangledown^\top), B_{k\ell}(x_\triangle)) = V_{ik}V_{j\ell}\exp\left(-\frac{1}{2}(x_\triangledown - x_\triangle)^\top 2\Lambda^{-1}(x_\triangledown - x_\triangle)\right),$$

and in particular, the marginal variance of any particular local Hessian element is

$$\operatorname{var}(B_{ij}(x_\triangledown)) = \operatorname{cov}(B_{ij}(x_\triangledown), B_{ij}(x_\triangledown)) = V_{ii}V_{jj}.$$

So the prior variance of element B_{ij} is $V_{ii}V_{jj}$, *not* V_{ij}, as one might think at first. Similarly, the length scale on which the elements change is not Λ, but $\Lambda/2$. So it is not possible to encode separate signal scales for the off-diagonal elements of the Hessian in this framework. They are determined entirely by the scales of the diagonal elements. Even so, if V is diagonal, then beliefs between any two different elements of B are independent.

Other kernels can of course be chosen; but it will become clear that an important practical requirement is the ability to efficiently integrate the kernel. This is feasible, though nontrivial, with the squared exponential kernel.

3.2 Line Integral Observations

For the Hessian $B(x)$ of a general function f, the quasi-Newton equation (4) is only a zeroth order approximation (a second-order approximation to f itself), assuming a constant Hessian everywhere. In our treatment, we will replace this approximate statement with its exact version: We observe the value of the *line integral* along the path $r^i : [0,1] \to \mathbb{R}^N$, $r^i(t) = x_{i-1} + t(x_i - x_{i-1})$,

$$Y_{ni} = \sum_m \int_{r^i_m} B_{nm}(x)\,\mathrm{d}x_m.$$

Note that, for scalar fields ϕ_i with $B_{im} = \nabla_m \phi_i$, such as the gradient $\phi_i = \nabla_i f$, it follows from the chain rule that (the following derivations again use the sum convention defined in Section 1.2)

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi_i(r^j(t)) = \nabla_m \phi_i(r^j(t))\frac{\partial r^j_m(t)}{\partial t} = B_{im}(r^j(t))\partial_t r^j_m(t).$$

Thus, our line integral obeys

$$\begin{aligned} Y_{ij} &= \int_{r^j} B_{im}(x)\,\mathrm{d}x_m = \int_0^1 B_{im}(r^j(t))\cdot\partial_t r^j_m(t)\,\mathrm{d}t \\ &= \int_0^1 \partial_t \phi_i(r^j(t))\,\mathrm{d}t = \phi_i(r^j(1)) - \phi_i(r^j(0)). \end{aligned} \tag{16}$$

This is the classic result that line integrals over the gradients of scalar fields are independent of the path taken, they only depend on the starting and end points of the path. In particular, our path satisfies $\partial_t r^j_m(t) = S_{jm}$ (its derivative is constant), and our line integral can be written as

$$\begin{aligned} Y_{ij} &= \int_0^1 B_{im}(r^j(t))S_{jm}\,\mathrm{d}t = \delta_{ik}\cdot S_{jm}\int_0^1 B_{km}(t^j)\,\mathrm{d}t^j, \\ \overrightarrow{Y} &= \left[I \otimes \left(S^\top \odot \int_t\right)\right]\overrightarrow{B} \equiv \mathfrak{S}_\triangleright \overrightarrow{B}. \end{aligned} \tag{17}$$

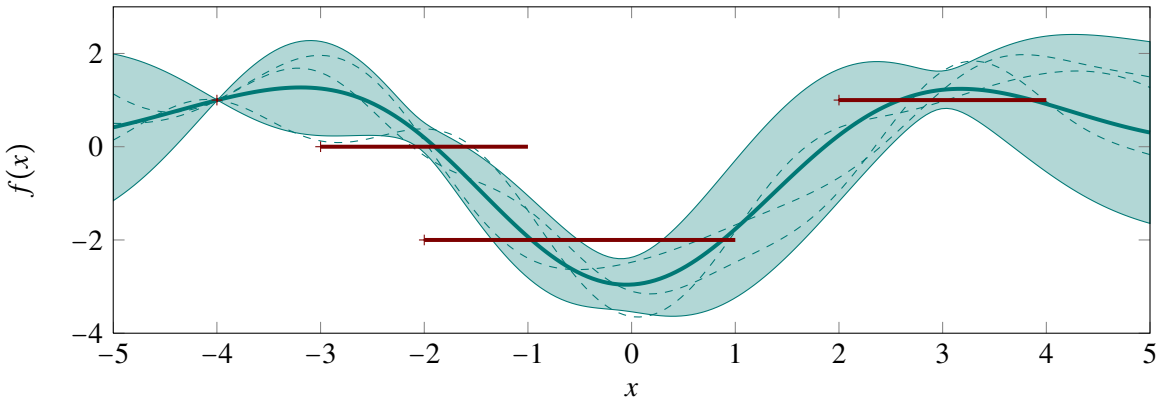

Figure 1: One-dimensional Gaussian process inference from integral observations (squared exponential kernel). Four observations, average values (integral value divided by length of integration region) and integration regions denoted by bars. Posterior mean in thick green, two standard deviations as shaded region, three samples as dashed lines. The left-most integral is over a very small region, which essentially reduces to the classical case of a local observation. Corresponding integrals over the mean, and each sample, are consistent with the integral observations.

where $\odot$ denotes the Hadamard, or element-wise, product $(a \odot b)_{k\ell} = a_{k\ell} b_{k\ell}$. In words: For every projection $S_{j:}$ of the *rows* of B, there are N one-dimensional *functions* $(Bs)_i(t^j) : \mathbb{R} \to \mathbb{R}$. Each of those functions are integrated from 0 to 1 (this is an affine projection onto the space of integrals over $[0,1]$). This amounts to taking *each component* $B_{ij}(t)$ of each projection and applying the integral-projection—hence the Hadamard product. We write the likelihood as

$$p(Y \,|\, B(x), \mathfrak{S}_{\triangleright}) = \lim_{\beta \to 0} \mathcal{N}\left[Y; \mathfrak{S}_{\triangleright}^{\top} \overrightarrow{B}, (k \otimes \beta I_M)\right],$$

using the linear operator $\mathfrak{S}_{\triangleright}$ defined in Equation (17). An interesting aspect to note is that, while path-independence holds for the ground-truth integrals of Equations (16), the prior covariance of Equation (15) does not encode this fact. The prior used here is more conservative than necessary, in the sense that it assigns nonzero probability mass on algebraically impossible functions, in exchange for lower computational cost. This is not unlike the aspects of parametric quasi-Newton methods discussed in Sections 2.1 and 2.2, where nonzero probability mass is assigned to the algebraically impossible case of non-symmetric Hessians. See Section 3.5 for more on this issue.

3.3 Gaussian Process Inference from Integral Observations

Because the Gaussian exponential family is closed under linear transformations, Gaussian process inference is analytic under any linear operator. Since integration is a linear operation, it is a corollary that Gaussian process inference is possible, in closed form, from integral observations. Nevertheless, this idea has only rarely been used in the literature (e.g., by Minka, 2000b). So we briefly digress here to introduce it in detail. Let there be a function $f(x) : \mathbb{R} \to \mathbb{R}$ (extension to multi-

variate functions is straightforward). Assume a Gaussian process prior, with mean function $\mu(x)$, covariance function (kernel) k. We observe, up to Gaussian noise, the value of a definite integral

$$y = \xi + \int_a^b f(x)\,\mathrm{d}x; \qquad \xi \sim \mathcal{N}(0, \sigma^2).$$

What is the posterior? We construct the answer from the finite-dimensional case, then take the Riemann limit. Consider observing the noisy weighted sum of N Gaussian variables, with weights δm_i:

$$y = \xi + \sum_i^N \delta m_i f_i \equiv \xi + m^\top f; \qquad p(f) = \mathcal{N}(\mu; K).$$

The posterior can be found as above, by "completing the square"

$$p(f\,|\,y) = \mathcal{N}(\Psi(K^{-1}\mu + \sigma^{-2} m y), \Psi)$$

with the covariance

$$\Psi = (K^{-1} + \sigma^{-2} m m^\top)^{-1} = K - \frac{K m m^\top K}{\sigma^2 + m^\top K m}.$$

Now consider the limit transition $N \to \infty$, such that the weights δm_i converge to a measure $m(x_i)\,\mathrm{d}x_i$ ($m = 1$ is a special case). We get

$$Km = K_{ij} m_j \to \int_a^b k(x_i, x_j)\,\mathrm{d}m(x_j) \quad \text{and}$$
$$m^\top K m = m_i K_{ij} m_j \to \iint_a^b k(x_i, x_j)\,\mathrm{d}m(x_i)\,\mathrm{d}m(x_j).$$

The mean has the form

$$\mu - \frac{K m m^\top \mu}{\sigma^2 + m^\top K m} + \sigma^{-2} K m \left(1 - \frac{m^\top K m}{\sigma^2 + m^\top K m}\right) y = \mu + K m \left(\frac{y - m^\top \mu}{\sigma^2 + m^\top K m}\right)$$

which, in the limit, transforms to

$$\mu + \frac{y - \int_a^b \mu(\tilde{x})\,\mathrm{d}m(\tilde{x})}{\sigma^2 + \iint_a^b k(x_i, x_j)\,\mathrm{d}m(x_i)\,\mathrm{d}m(x_j)} \int_a^b k(x_i, x_j)\,\mathrm{d}m(x_i).$$

Figure 1 gives a toy 1D example for intuition.

3.4 Posterior on Hessians

Using an argument entirely analogous to that of Section 2, we find that the primal posterior after M observations has mean function

$$\begin{aligned}
&\overrightarrow{B}_\triangleright(x_\forall) \\
&= \overrightarrow{B}_0(x_\forall) + (\Sigma \mathfrak{S}_\triangleright)(x_\forall)(K + (k \otimes \beta I))^{-1}(Y - \mathfrak{S}_\triangleright^\top \overrightarrow{B}_0) \\
&= \overrightarrow{B}_0 + \left[k \otimes k\left(S \odot \int_t\right)\right]\left(k \otimes \left(S \odot \int\right)^\top k\left(S \odot \int\right) + k \otimes \beta I\right)^{-1} (Y - \mathfrak{S}_\triangleright^\top \overrightarrow{B}_0).
\end{aligned}$$

The terms of this equation can be further identified using the Gram matrix

$$\begin{aligned}\left(S \odot \int\right)^{\top} k \left(S \odot \int\right)\Big|_{j\ell} &= \int_0^1 \mathrm{d}t^j \int_0^1 \mathrm{d}t^\ell S_k^j k_{km}(x_\forall(t^j), x_\triangle(t^\ell)) S_m^\ell \\ &= S_{jk}\left[\iint_0^1 k_{km}(x_\forall(t^j), x_\triangle(t^\ell))\,\mathrm{d}t^j\,\mathrm{d}t^\ell\right] S_{m\ell} \\ &\equiv \mathfrak{K} \in \mathbb{R}^{M\times M},\end{aligned} \tag{18}$$

the integrated kernel map

$$\begin{aligned}k\left(S \odot \int\right)\Big|_{kj}(x_\forall) &= \int_0^1 k_{km}(x_\forall, x_\triangle(t^j))\,\mathrm{d}t^j S_{jm} \\ &\equiv \mathfrak{k}(x_\forall) \in \{\mathbb{R}^N \to \mathbb{R}^{N\times M}\},\end{aligned} \tag{19}$$

and the integrated mean function

$$\mathfrak{S}_{\triangleright}^{\top} \overrightarrow{B}_0\Big|_{mk} = S_{jk} \int_0^1 B^0_{mj}(x(t^k))\,\mathrm{d}t^k \equiv \mathfrak{B} \in \mathbb{R}^{N\times M}. \tag{20}$$

These objects are homologous to concepts in canonical Gaussian process inference: $\mathfrak{B}_{0,nm}$ is the n-th mean prediction along the m-th line integral observation. $\mathfrak{k}_{nm}(x_\forall)$ is the covariance between the n-th column of the Hessian at location $x_\forall$ and the m-th line-integral observation. $\mathfrak{K}_{pq}$ is the covariance between the p-th and q-th line integral observations. The derivations for the covariance are similar and contain the same terms. Together with the dual observation, we arrive at a posterior, which has mean and covariance functions

$$\begin{aligned}B_\diamond(x_\forall) = B_0(x_\forall) &+ (Y - \mathfrak{B}_0)\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\forall) + \mathfrak{k}(x_\forall)\mathfrak{K}^{-1}(Y - \mathfrak{B}_0)^{\top} \\ &- \mathfrak{k}(x_\forall)\mathfrak{K}^{-1}S^{\top}(Y - \mathfrak{B}_0)\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\forall), \\ \Sigma_\diamond(x_\forall, x_\triangle) = \left[k(x_\forall^{\top}, x_\triangle^{\top}) - \mathfrak{k}(x_\forall^{\top})\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\triangle)\right] &\otimes \left[k(x_\forall, x_\triangle) - \mathfrak{k}(x_\forall)\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\triangle)\right].\end{aligned}$$

The actual numerical realisation of this nonparametric method involves relatively tedious algebraic derivations, which can be found in Appendix A.

An important aspect is that, because k is a positive definite kernel, unless two observations are exactly identical, $\mathfrak{K}$ has full rank M (the number of function evaluations), even if several observations take place within one shared 1-dimensional subspace. So it is possible to make full use of *all* function evaluations made during line searches, not just the last one, as in the parametric setting of existing quasi-Newton methods. Figure 2 uses another toy setting to give an intuition for why this matters. Just as in Section 2.3, it is clear that the posterior mean is not in general a symmetric matrix. So we project into the space of symmetric matrices using the arguments surrounding Equation (12).

A downside is that evaluating the mean function involves finding the inverse of $\mathfrak{K}$, at cost $\mathcal{O}(M^3)$. Two aspects of numerical optimization make this issue less problematic than one might think. First, solving an optimization problem takes finite time, often just a few hundred evaluations; so the cubic cost in M is often manageable. Where it is not, note that, because optimization proceeds along a trajectory through the parameter space, old observations tend to have low covariance with the Hessian at the current location, and thus a small effect on the local mean estimate. So they can often simply be ignored. The simplest possible way to do so is to just throw away all observations

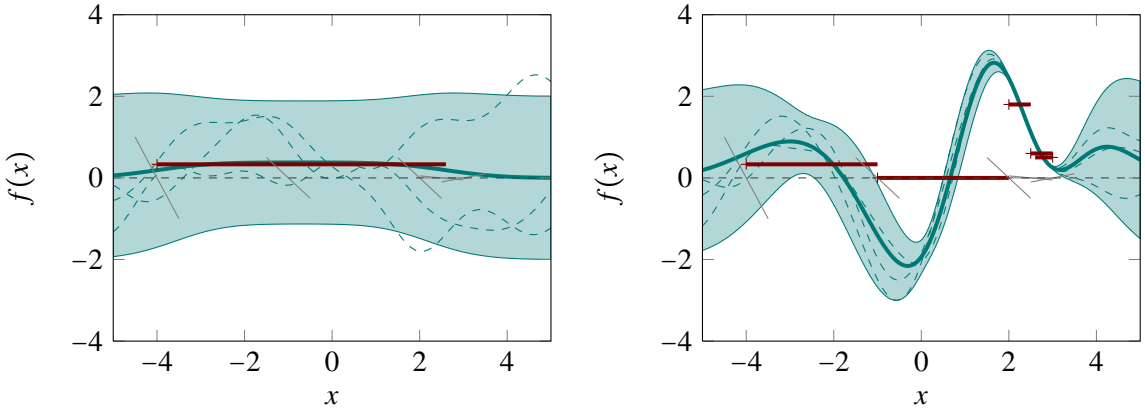

Figure 2: Simulated line search, a toy problem to elucidate why it helps to use all line search observations instead of only the first and last ones. Observations at locations $X = [-4; -1; 2; 2.5; 3; 2.6]$, observed 1D Gradients of $\nabla f(X) = [-2; -1; -1; -0.1; 0.2; -0.001]$. **Left:** Traditional inference based on only the first and last observations. **Right:** Our nonparametric model can use all observations. Gaussian process posterior with thick mean and two standard deviations marginal variance as shaded region, as well as three samples as dashed lines. Effective observations $y_i/(x_i - x_{i-1})$ as bars. Gradient values as thin lines on the abscissa for intuition. The posterior from all observations captures much more structure, and in particular a different mean estimate at the end of the line search ($x = 2.6$), where its value defines the next search direction.

older than some memory bound M_0. This is the approach of the L-BFGS method (Nocedal, 1980). Since the regression framework quantifies the contribution of each observation to the prediction, in the vector $\natural\mathfrak{K}^{-1}$, we can also use the relative sizes of these elements to order past observations and discard those ranked below M_0.

3.5 Diversion: Naïve Gaussian Regression is Too Costly

The discussion in Section 2 established that, with a few caveats (Section 2.2), quasi-Newton methods are Gaussian regressors; and we then extended to nonparametric Gaussian process inference. Importantly, the prior from Section 3.1 is over the elements of the Hessian, and gradient observations are integrals of this function. One may wonder why we did not just start with a Gaussian process prior on the objective function f and used observations of the gradient to infer the Hessian directly from there. This is possible because differentiation, like integration, is a linear operation: Under a Gaussian process prior on f with kernel k^f and mean function μ^f, the mean function of the prior belief over the gradient is $\mu_{\nabla f} = \nabla \mu^f$, and the covariance between elements of ∇f at two

different points $x^\triangledown$ and $x^\triangle$ is (Rasmussen and Williams, 2006, §9.4)

$$\operatorname{cov}\left(\frac{\partial f(x^\triangledown)}{\partial x_i^\triangle}, \frac{\partial f(x^\triangledown)}{\partial x_j^\triangledown}\right) = \frac{\partial^2 k(x^\triangledown, x^\triangle)}{\partial x_i^\triangle, \partial x_j^\triangledown}, \qquad \text{which, for an SE kernel, is} \tag{21}$$
$$= \left(\frac{1}{\lambda_j^2}\delta_{ij} + \frac{(x_i^\triangle - x_i^\triangledown)(x_j^\triangle - x_j^\triangledown)}{\lambda_i^2\lambda_j^2}\right) k_{SE}(x^\triangle, x^\triangledown).$$

The covariance between elements of the Hessian and elements of the gradient is

$$\operatorname{cov}\left(\frac{\partial^2 f(x^\triangledown)}{\partial x_i^\triangle \, \mathrm{d}x_k^\triangle}, \frac{\partial f(x^\triangledown)}{\partial x_j^\triangledown}\right) = \frac{\partial^2 k(x^\triangledown, x^\triangle)}{\partial x_i^\triangle, \partial x_j^\triangledown}, \qquad \text{which, for an SE kernel, is}$$
$$= \left(\frac{\delta_{ik}(x_j^\triangle - x_j^\triangledown) + \delta_{jk}(x_i^\triangle - x_i^\triangledown)}{\lambda_i^2\lambda_j^2} - \frac{(x_k^\triangle - x_k^\triangledown)}{\lambda_k^2}\right) k_{SE}(x^\triangle, x^\triangledown).$$

So, given observations of the gradient at M points x^m, we can evaluate the mean over the elements of the Hessian $B(x^*)$ as

$$\hat{B}_{ik}(x^*) = \mu_B(x^*) + \operatorname{cov}(B_{ik}(x^*), \nabla_j f(x^m)) K^{-1}_{(jm)(\ell q)} (\nabla_\ell f(x^q) - \mu_{\nabla_\ell f}(x^q)),$$

with a Gram matrix $K \in \mathbb{R}^{MN \times MN}$ of elements $K_{(jm)(\ell q)} = \operatorname{cov}(\nabla_j f(x^m), \nabla_\ell f(x^q))$. From Equation (21) we see that this Gram matrix has specific structure, so it might be possible to construct its inverse faster than in $\mathcal{O}(M^3N^3)$. But even then, this scheme would only provide a belief over the *elements* of B. Since Newton's method requires the *inverse* of B, this mean prediction would still have to be inverted, at cost $\mathcal{O}(N^3)$. This would defeat the point of a quasi-Newton method: constructing a low-rank estimate of the Hessian, and thus a fast estimate of its inverse. If N is small enough to allow for general (cubic) inversion of $\hat{B}$, we might as well just calculate the true Hessian of f and invert that instead. So quasi-Newton methods are not "just" standard Gaussian regression on Hessians. Their key advantage stems from the weaker prior assumptions, as discussed in Sections 2.1 and 2.2, which allow the construction of a low-rank estimate.

4. Experiments

The calculations required by nonparametric quasi-Newton algorithm using the squared-exponential kernel involve exponential functions, error functions, and numerical integrals (see Appendix A for details). A side-effect of these is that this algorithm has slightly lower numerical precision than its predecessors. This issue becomes clear when minimizing quadratic functions (Figure 3), whose constant Hessian voids the modeling advantage of the nonparametric method:[2] The nonparametric algorithm behaves more regularly initially, but towards the end of the optimization process the numerical conditioning of the kernel calculations begins to play a role, offering an advantage to the better conditioned older methods. In real, non-quadratic optimization problems, however, this problem only arises close to the end of optimization, when the algorithm is very close to the optimum. In

2. This is only a diagnostic example. Quadratic functions, whose optimization amounts to solving a positive-definite linear program, are not a realistic use-case for quasi-Newton methods, parametric or not. Specialised methods, like the method of conjugate gradients (Hestenes and Stiefel, 1952), or plain Cholesky decomposition for low-dimensional cases, are better suited for this simple setting.

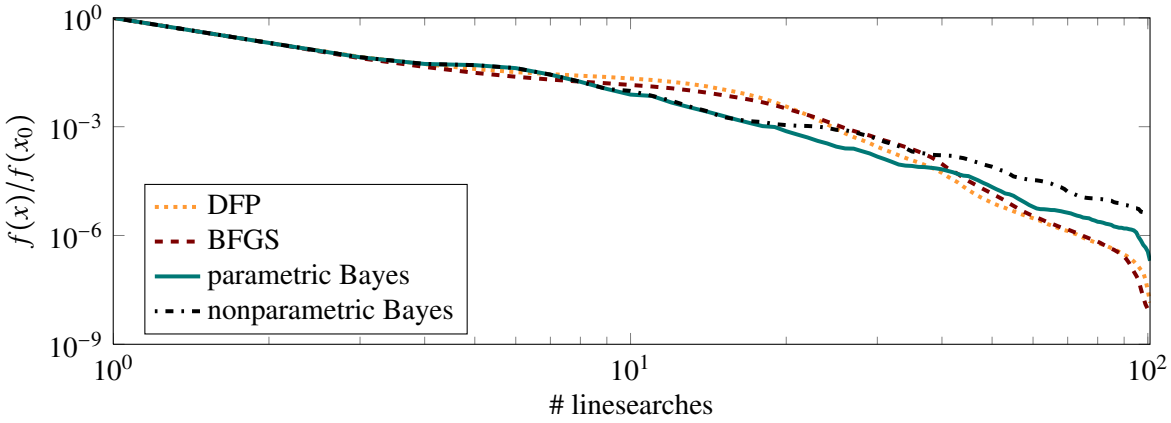

Figure 3: Minimization of a 100-dimensional quadratic. All algorithms shared the same line search method. Averages over 20 sampled problems (see text for details). The two dashed lines in this log-log plot mark linear and quadratic convergence. The Bayesian algorithms converge more regularly and faster initially, but suffer from bad numerical conditioning toward the end of the optimization.

this small region, a local quadratic approximation is valid and the Hessian is essentially constant. In our practical implementation, we thus check for convergence, then pass the learned inverse Hessian to the better conditioned BFGS for the final few steps, in which the learned Hessian barely changes.

For intuition, Figure 4 shows results from a popular two-dimensional test problem—Rosenbrock's polynomial. The plot also shows the mean belief on one element of the Hessian. The availability of this explicit estimate for the entire function is an additional benefit of the nonparametric method.

In problems where the Hessian is not constant everywhere, the nonparametric Bayesian optimizer can sometimes offer drastic advantages over the classical alternatives. Figure 5, left, shows averages of experiments on a 200-dimensional domain. The objective functions is a prior over hyperparameters of a Gaussian process regressor: the logarithm of products of Gamma distributions, with different parameters for each dimension. The right part of the figure shows that the performance advantage is not always so drastic. It was gathered on the corresponding posterior after the addition of 10 datapoints per problem. This makes the objective function less regular, meaning that the optimal Newton path to the minimum has more complex shape, and more line searches are necessary to converge to the minimum.

Figure 6 shows performance on a set of low-dimensional but challenging set of problems: Functions sampled from a Gaussian process with quadratic prior mean, after conditioning on 10 observations of the function's Hessian (drawn separately from a Wishart distribution, to ensure positive definiteness). In all experiments, however, the Bayesian algorithm performs at least as good, and regularly better than its classical competitors. For numerical optimization, even performance gains of a few percent are valuable, because optimization is such a widely encountered problem. Speeding up quasi-Newton methods by 10% means speeding up large parts of machine learning by that amount. Our experiments show that, at least in some cases, the new algorithm offers improvements much beyond that.

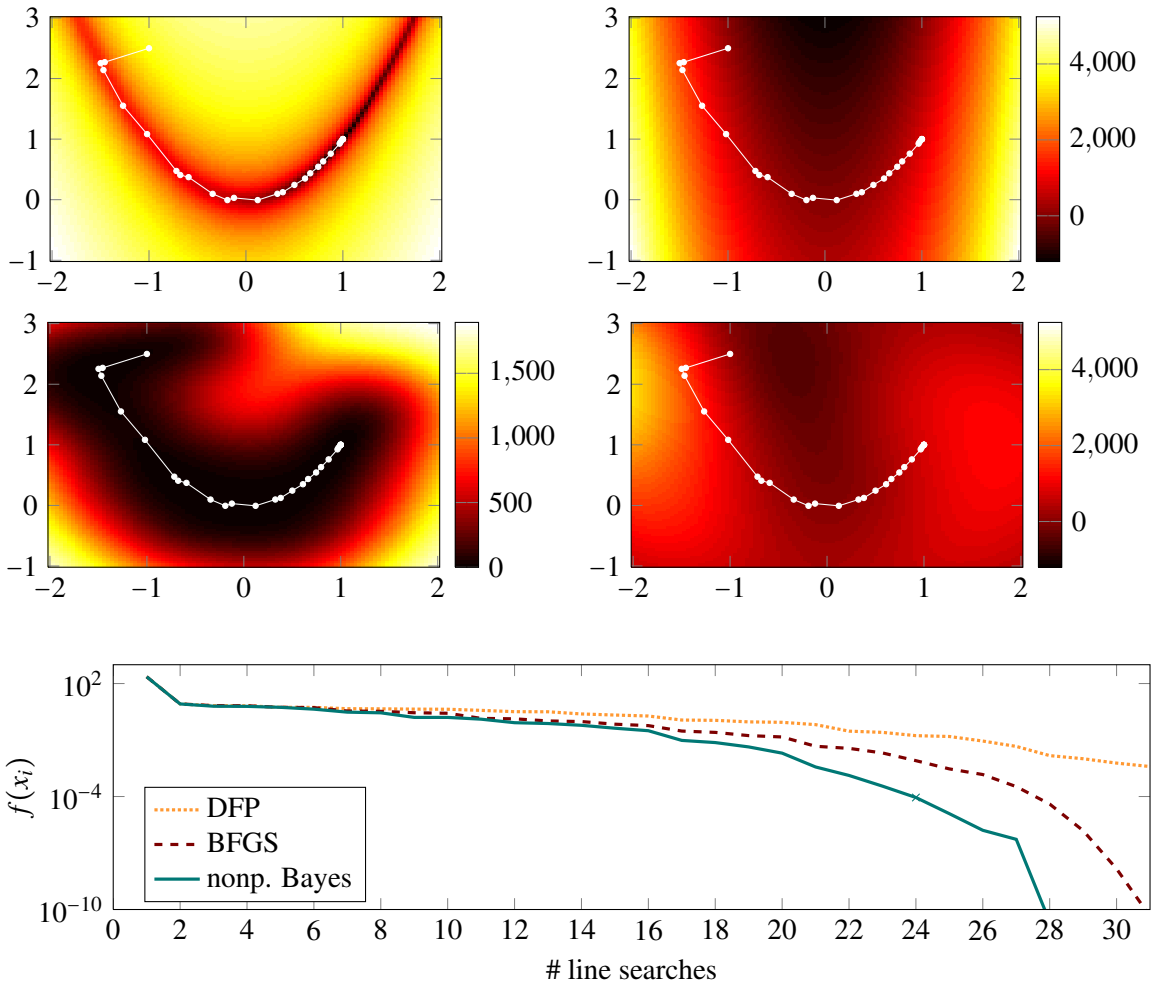

Figure 4: Minimizing Rosenbrock's polynomial, a non-convex function with unique minimum at (1,1). All algorithms start from (-1,2.5). **Top left:** Function values, line search trajectory of the Bayesian algorithm in white. **Top Right:** True value of the (1,1) element of the Hessian (other elements have less interesting structure). **Middle Row:** Two times marginal posterior standard deviation (a.k.a. posterior uncertainty, left) and mean estimate (right) of the Bayesian regressor. Comparing the top right and middle right plots shows good agreement in the regions visited by the algorithm. **Bottom:** function value as function of the number of line searches. The cross after 24 line searches marks the point where the Bayesian method switches to a local parametric model for numerical stability.

4.1 Cost

As pointed out above, the computational complexity of this algorithm, given a diagonal prior mean, is $\mathcal{O}(NM + M^3)$ per update of the search direction, where M is the number of function evaluations used to build the model (which can be controlled ad hoc within the algorithm by excluding redun-

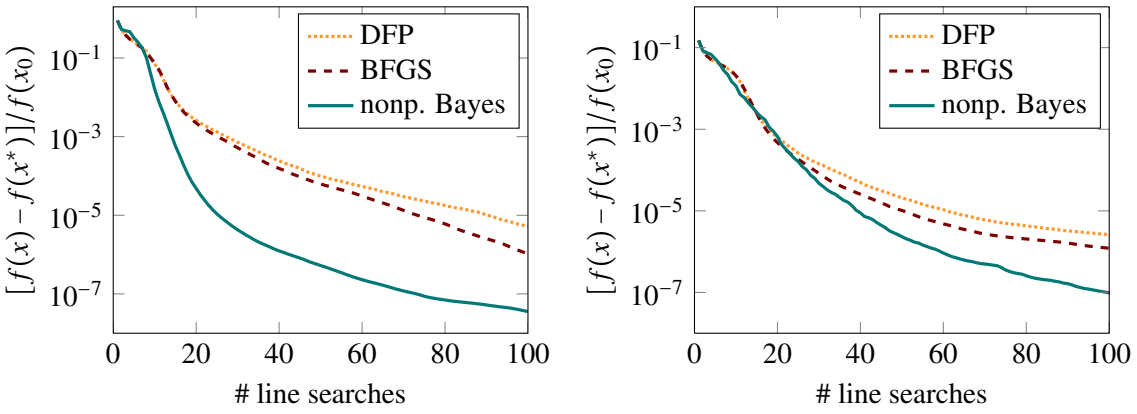

Figure 5: **Left:** Minimizing the 200-dimensional (Gamma) prior over the hyperparameters of a Gaussian process regression modell. **Right:** Minimizing the corresponding *posterior* after the addition of 10 datapoints sampled from the correct model. The datapoints create a more complicated optimization problem in which line searches tend to be shorter, thus reducing the advantage of the Bayesian method gained from superior Hessian estimates. Averages over 20 sampled problems; plotted is the relative distance from initial function value (shared by all algorithms) to the minimum, as a function of the number of line searches (all algorithms use the same line search method).

dant or irrelevant evaluations). This compares to $\mathcal{O}(NM)$ for the corresponding cases of DFP and BFGS. Although the overhead created by the squared-exponential integrals is nontrivial, we found the computational demands of our implementation manageable: In our experiments, the cost of constructing and inverting the matrix $\mathfrak{K}$ was negligible, and could, in very time-sensitive settings, be further reduced by a more efficient implementation.

5. Outlook

In this paper we primarily focused on a better understanding for quasi-Newton methods. For an intuition on the potential of Bayesian formulations of numerical optimization, apart from the new nonparametric algorithm derived in Section 3 and tested in Section 4, consider some potential for future work: Perhaps the most obvious insight is that Gaussian process regression is trivial to extend to noisy evaluations. An upcoming conference paper (Hennig, 2013) will study how this can be used to construct optimizers robust to noise. Repeated integration, and non-Gaussian likelihoods in combination with approximate inference, may allow optimization without gradients, and from only gradient sign observations, respectively. Structured and hierarchical priors are a third direction, offering new avenues for optimization of very high-dimensional functions.

6. Conclusion

We have shown that the most popular quasi-Newton algorithms can be interpreted as approximations to Bayesian regression under Gaussian and other priors. This deepens our understanding of these algorithms. In particular, it emerged that symmetry in the estimators of SR1, PSB, DFP and BFGS,

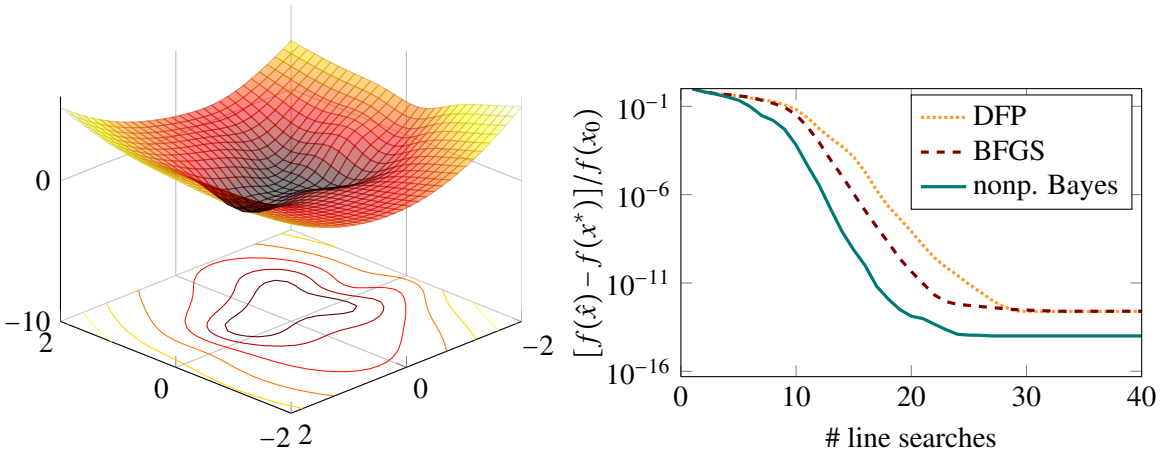

Figure 6: Minimizing randomly generated 4-dimensional analytic functions. **Left:** For illustration. One slice through the $(x, y, 0, 0)$ plane of one of the sampled functions. Neither starting point of the search nor the found optimum lie within this slice, and are thus not shown. **Right:** function values achieved by three numerical optimizers as a function of the number of line searches. All algorithms shared the same line search routine. Plotted is the difference between best function value achieved by any of the optimizers for each function, normalized by the initial function value (which is identical for all algorithms). The lines are averages of the logarithmic values from 10 iid. experiments.

and positive definiteness in those of DFP and BFGS, are encoded in approximate ways which do not capture all available prior information but allow for low computational cost.

As a parallel result, our analysis also gives rise to a new class of Bayesian nonparametric quasi-Newton algorithms. These use a kernel model to learn from all observations in each line-search, explicitly track uncertainty, and thus achieve faster convergence towards the true Hessian. While the new methods are not trivial to understand and implement, their computational cost lies within a constant of that of their predecessors. Our research implementation is available at `http://www.probabilistic-optimization.org/Newton.html`.

Acknowledgments

We would like to thank Christian Schuler, Tom Minka, and Carl Rasmussen and the anonymous reviewers for helpful comments. MK's work is supported by a grant from Microsoft Research Ltd.

Appendix A. Numerical Implementation

As mentioned above, for a concrete implementation, we chose to use the squared exponential kernel (15), and a constant mean function assigning $B_0(x_\forall) = I$ everywhere. It is another advantage of the Bayesian formulation that prior assumptions are directly accessible for analysis: The squared expo-

nential prior amounts to the assumption that the elements of the Hessian vary independently over the parameter space, on one unique set of length-scales Λ. Multiple length scales could be modeled using sums of kernels, but our implementation does not currently offer this option. Changing the length scales Λ amounts to a form of pre-conditioning. The fact that this can be done automatically using methods from machine learning is another advantage of a Bayesian formulation. A naïve approach for such an optimization would be to optimize the hyperparameters by type-II maximum likelihood, as is often done in standard Gaussian process regression. Since this amounts to an optimization problem itself, though, one might hope to find closed form estimators. We will not dwell further on this issue here, leaving it for future work.

A numerical challenge in the implementation arises from the required integrals over squared exponentials. Of the three objects in Equations (18), (19), and (20) only the last one, $\mathfrak{B}$, is truly straightforward, thanks to our choice of constant mean function. The other two will be derived in this section. For this purpose, it is helpful to use an explicit notation for individual line searches: We change the index set from m to (jh): Let observation y_m have been taken as the h-th observation of line search number j. If the line search proceeded along unit direction e_j and started from x_{0j}, then the h-th observation was the difference between the gradients at locations $x_{0j} + (\eta_h - \nu_h)e_j$ and $x_{0j} + \nu_h e_j$.

A.1 $\mathfrak{k}$

The elements of the $N \times M$ matrix $\mathfrak{k}(x_\forall)$ are, (the ellipses are placeholders for the second, analogous part of quadratic forms)

$$\begin{aligned}
\mathfrak{k}^j_{nh}(x_\forall) &= (\eta_h - \nu_h) V_{nm} e^j_m \int_0^1 \exp\left[-\frac{1}{2}(x_\forall - (\nu_h e_j + x_{0j}) - (\eta_h - \nu_{h-1}) t e_j)^\top \Lambda^{-1} \dots \right] \mathrm{d}t \\
&= (Ve^j)_n \exp\left(-\frac{c - b^2/a^2}{2} \right) \frac{1}{a} \int_b^{(\eta_h - \nu_h)a + b/a} \exp\left(-\frac{u^2}{2} \right) \mathrm{d}u \\
&= (Ve^j)_n \exp\left(-\frac{c - b^2/a^2}{2} \right) \sqrt{\frac{\pi}{2a^2}} \left[\operatorname{erf}\left(\frac{(\eta_h - \nu_h)a^2 + b}{\sqrt{2a^2}} \right) - \operatorname{erf}\left(\frac{b}{\sqrt{2a^2}} \right) \right],
\end{aligned}$$

with

$$\begin{aligned}
a &= \sqrt{e_j^\top \Lambda^{-1} e_j} \\
b &= e_j^\top \Lambda^{-1} (x_{0j} + \nu_h e_j - x_\forall) \\
c &= x_\forall^\top \Lambda^{-1} x_\forall - 2 x_\forall \Lambda^{-1} (x_{0j} + \nu_h e_j) + (x_{0j} + \nu_h e_j)^\top \Lambda^{-1} (x_{0j} + \nu_h e_j).
\end{aligned}$$

This involves the error function, for which good double-precision approximations are widely available.

A.2 $\mathfrak{K}$

The $M \times M$ matrix $\mathfrak{K}$ has two types of elements. Along its block diagonal lie covariance between observations collected as part of the same line search. These have the form

$$\begin{aligned}\mathfrak{K}^{ii}_{hk} &= (\eta_h - \nu_h)(\eta_k - \nu_k)(e_i^\top V e_i)\theta^2 \\ &\quad \iint_0^1 \exp\left[-\frac{1}{2}[(\eta_h - \nu_h)t_h e_i + \nu_h e_i + x_{0i} - (\alpha_k - \nu_k)t_k e_i - \nu_k e_i - x_{0i}]^\top \Lambda^{-1}[\dots] \right] \mathrm{d}t_h \, \mathrm{d}t_k \\ &= e_i^\top V e_i \theta^2 \int_{\nu_h}^{\eta_h} \int_{\nu_k}^{\alpha_k} \exp\left[-\frac{(u_h - u_k)^2}{2\sigma_i^2} \right] \mathrm{d}u_h \, \mathrm{d}u_k.\end{aligned}$$

So these terms are double integrals over a one-dimensional squared exponential. Such integrals can be integrated by parts, leading to an analytic expression that only involves error functions and exponential functions (Peltonen, 2012).

The most challenging calculations involve elements of $\mathfrak{K}$ describing the covariance between observations made along different line search directions. We make use, once more, of the closure of the Gaussian exponential family under linear maps, to write

$$\begin{aligned}\mathfrak{K}^{ij}_{hk} =& (\eta_h - \nu_h)(\eta_k - \nu_k) e_j^\top V e_i \\ & \iint_0^1 \exp\left[-\frac{1}{2} \begin{pmatrix} (\eta_h - \nu_h)t_h \\ (\eta_k - \nu_k)t_k \\ 1 \end{pmatrix}^\top \begin{pmatrix} e_j \\ -e_i \\ \nu_h e_j + x_{0j} - \nu_k e_i - x_{0i} \end{pmatrix}^\top \Lambda^{-1} \dots \right] \mathrm{d}t_h \, \mathrm{d}t_k \\ =& \frac{2\pi e_j^\top V e_i \exp[-\frac{1}{2}(c - b^\top A^{-1} b)]}{\sqrt{(1-\rho^2)A_{hh}A_{kk}}} \\ & \left[\Phi(u_{hf}, u_{kf}, \rho) + \Phi(u_{hi}, u_{ki}, \rho) - \Phi(u_{hi}, u_{kf}, \rho) - \Phi(u_{hf}, u_{ki}, \rho) \right]\end{aligned}$$

with the bivariate Gaussian CDF

$$\Phi(b_1, b_2, \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} \exp[-(x^2 - 2\rho xy + y^2)/2(1-\rho^2)] \, \mathrm{d}x \, \mathrm{d}y$$

and

$$\begin{aligned} A^{-1} &= \begin{pmatrix} e_j^\top \Lambda^{-1} e_j & -e_j^\top \Lambda^{-1} e_i \\ -e_j^\top \Lambda^{-1} e_i & e_i^\top \Lambda^{-1} e_i \end{pmatrix}^{-1} = \frac{\begin{pmatrix} e_i^\top \Lambda^{-1} e_i & e_j^\top \Lambda^{-1} e_i \\ e_j^\top \Lambda^{-1} e_i & e_j^\top \Lambda^{-1} e_j \end{pmatrix}}{e_j^\top \Lambda^{-1} e_j e_i^\top \Lambda^{-1} e_i - (e_j^\top \Lambda^{-1} e_i)^2}, \\ b &= \begin{pmatrix} e_j^\top \Lambda^{-1}(\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i}) \\ -e_i^\top \Lambda^{-1}(\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i}) \end{pmatrix}, \\ c &= (\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i})^\top \Lambda^{-1}(\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i}),\end{aligned}$$

as well as

$$\begin{aligned} \rho &= \frac{e_j^\top \Lambda^{-1} e_i}{\sqrt{e_j^\top \Lambda^{-1} e_j e_i^\top \Lambda^{-1} e_i}}, \\ u_i &= \sqrt{1-\rho^2} \, \mathrm{diag}(\sqrt{[A_{hh}, A_{kk}]}) A^{-1} b, \\ u_f &= \sqrt{1-\rho^2} \, \mathrm{diag}(\sqrt{[A_{hh}, A_{kk}]}) \left[\begin{pmatrix} \eta_h - \nu_h \\ \eta_k - \nu_k \end{pmatrix} + A^{-1} b \right].\end{aligned}$$

Just like in the univariate case, bivariate Gaussian CDFs are not analytic. But single and double precision numerical approximations at acceptable computational cost exist (Genz, 2004).

From Sec. 1, recall that updating the search direction requires the *inverse* of B. Explicit inversion costs $\mathcal{O}(N^3)$, but the inverse can be constructed analytically, from the matrix inversion lemma, in $\mathcal{O}(N^2+NM+M^3)$. Using an argument largely analogous to the derivation of the L-BFGS algorithm (Nocedal, 1980) a diagonal prior mean B_0 lowers cost to $\mathcal{O}(NM+M^3)$, linear in N. Just like L-BFGS, the nonparametric method is thus applicable to problems of even very high dimensionality.

References

S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ Press, 2004.

C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19 (92):577–593, 1965.

C.G. Broyden. Quasi-Newton methods and their application to function minimization. *Math. Comp.*, 21(368):45, 1967.

C.G. Broyden. A new double-rank minimization algorithm. *Notices American Math. Soc*, 16:670, 1969.

C.G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76, 1970.

W.C. Davidon. Variable metric method for minimization. Technical report, Argonne National Laboratories, Ill., 1959.

J.E. Jr Dennis and J.J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, pages 46–89, 1977.

R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317, 1970.

R. Fletcher and M.J.D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163–168, 1963.

A. Genz. Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14(3):251–260, 2004.

D. Goldfarb. A family of variable metric updates derived by variational means. *Math. Comp.*, 24 (109):23–26, 1970.

J. Greenstadt. Variations on variable-metric methods. *Math. Comp*, 24:1–22, 1970.

P. Hennig. Fast probabilistic optimization from noisy gradients. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

H. Lütkepohl. *Handbook of Matrices*. Wiley, 1996.

T.P. Minka. Old an new linear algebra useful for statistics. Technical report, MIT Media Lab Note, 2000a.

T.P. Minka. Deriving quadrature rules from Gaussian processes. Technical report, Statistics Department, Carnegie Mellon University, 2000b.

J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comp.*, 35(151):773–782, 1980.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, 1999.

J.D. Pearson. Variable metric methods of minimisation. *The Computer Journal*, 12(2):171–178, 1969.

J. Peltonen. Personal communication, 2012.

M.J.D. Powell. A new algorithm for unconstrained optimization. In O. L. Mangasarian and K. Ritter, editors, *Nonlinear Programming*. AP, 1970.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

R.P. Savage. The space of positive definite matrices and Gromov's invariant. *Transactions of the AMS*, 274(1):239–263, November 1982.

D.F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.*, 24 (111):647–656, 1970.