Supporting Theory and Data Analysis for "Long Range Search for Maximum Likelihood in Exponential Families"

By

Saisuke Okabayashi and Charles J. Geyer

Technical Report No. 686
School of Statistics
University of Minnesota
July 22, 2011

Abstract: Exponential families are often used to model data sets with complex dependence. Maximum likelihood estimators (MLE) can be difficult to estimate when the likelihood is expensive to compute. Markov chain Monte Carlo (MCMC) methods based on the MCMC-MLE algorithm in Geyer and Thompson (1992) are guaranteed to converge in theory under certain conditions when starting from any value, but in practice such an algorithm may labor to converge when given a poor starting value. We present a simple line search algorithm to find the MLE of a regular exponential family when the MLE exists and is unique. The algorithm can be started from any initial value and avoids the trial and error experimentation associated with calibrating algorithms like stochastic approximation. Unlike many optimization algorithms, this approach utilizes first derivative information only, evaluating neither the likelihood function itself nor derivatives of higher order than first. We show convergence of the algorithm for the case where the gradient can be calculated exactly. When it cannot, it has a particularly convenient form that is easily estimable with MCMC, making the algorithm still useful to a practitioner.

Keywords and phrases: Markov chain Monte Carlo, exponential families, Potts, Ising, exponential random graph, stochastic approximation.

1. Introduction

Exponential families are commonly used to model phenomena with dependent structure, where the outcomes of the response variable of interest are in fact dependent on one another. For example, the Ising model (Ising, 1925; Potts, 1952) is an exponential family model that has been used to model ferromagnetism. A realized sample from this model is depicted in Figure 1, where neighboring pixels (representing atoms in a crystal lattice) are more likely to have the same color. We explore this model further in Section 5.2. Other examples of phenomena with dependent structure modeled with exponential families include plant ecology (Besag, 1974, 1975), friendship networks (Goodreau, 2007; Goodreau, Kitts and Morris, 2009; Wasserman and Pattison, 1996), protein-protein interaction networks (Saul and Filkov, 2007), and the lifetime fitness of plants (Shaw et al., 2008).

The appeal of exponential families in these settings stems from their simplicity and maximum entropy property (Geyer and Thompson, 1992; Jaynes, 1978). By choosing statistics of interest on the data, one fully specifies a model that gives the most reasonable inference possible derived solely from those statistics. Furthermore, exponential families have been well-studied (Barndorff-Nielsen, 1978; Brown, 1986) and utilized over the decades and have desirable properties such as a strictly concave likelihood function.

1.1. Parameter estimation methods in exponential families

Calculating the maximum likelihood estimators (MLE) for exponential families when dependence is complex, however, remains a challenging problem because the likelihood function may be computationally infeasible. In particular, the

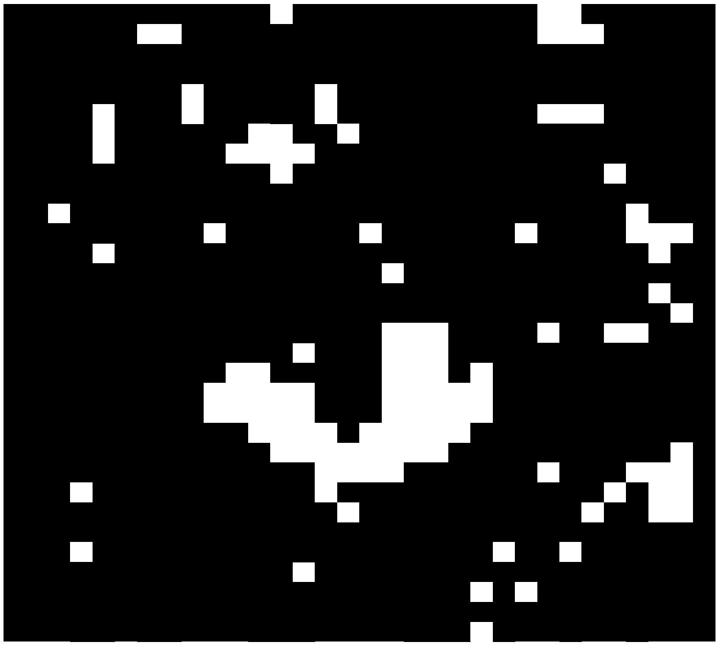

Fig 1. *A realized sample from an Ising model on a* 32×32 *lattice with* $\eta = \left(0, \log(1+\sqrt{2})\right)^T$. *This value of* η *corresponds to the phase transition point, where the sample images are mostly one color with small but significant portions of the other color. There is no preference for the dominant color to be white or black.*

form of the likelihood is determined by the chosen statistics up to a normalizing constant, but this normalizing constant may involve a summation over an astronomical number of terms. Evaluating the likelihood function—let alone maximizing it—presents a significant challenge.

Two commonly used parameter estimation methods to circumvent this issue in exponential families are the *pseudo-likelihood* approach (Besag, 1975; Okabayashi, Johnson and Geyer, 2011; Strauss and Ikeda, 1990), which finds parameter values that maximize the pseudo-likelihood function, and the *Markov chain Monte Carlo maximum likelihood estimate* (MCMC-MLE) approach (Geyer, 1994; Geyer and Thompson, 1992), which uses MCMC to approximate the log likelihood so that it can subsequently be maximized. The pseudo-likelihood approach is computationally expedient, but has been shown to produce unreliable results when dependence is strong (Geyer and Thompson, 1992; van Duijn, Gile and Handcock, 2009).

The MCMC-MLE approach is theoretically guaranteed to converge to the MLE if it exists and is the default algorithm in software packages such as `statnet` (Handcock et al., 2003) in the R platform for network models. However, this approach has been shown in practice to be sensitive to initial parameter values when used without the trust region methodology recommended in Geyer and Thompson (1992), and the algorithm may require many iterations and enormous (sometimes infeasibly large) Monte Carlo sample sizes when the starting value is far from the MLE (Hunter et al., 2008). Improvement to the MCMC-MLE approach is an active area of research (Hummel, Hunter and Handcock, 2010).

Variations on the Robbins-Monro *stochastic approximation* (SA) algorithm (Robbins and Monro, 1951) have been applied to find the MLE similar contexts: Gu and Zhu (2001); Moyeed and Baddeley (1991); Younes (1988, 1989) applied MCMC stochastic approximation to spatial models and Snijders (2002) to social network models (exponential random graph models). SA procedures for finding the MLE of a parameter η generate iterated estimates η_k to find the root of a gradient function $h(\eta)$:

$$\eta_{k+1} = \eta_k + \alpha_k U_k, \tag{1}$$

where α_k is a step size and is typically a member of a decreasing sequence of positive numbers, and U_k is a random variable from the distribution specified by η_k that noisily estimates the gradient function $h(\eta_k)$.

Restrictive conditions are required of α_k and U_k to establish convergence of the sequence η_k. In Robbins-Monro SA (Robbins and Monro, 1951), the step size α_k must be a sequence of positive constants that satisfies

$$\sum \alpha_k^2 < \infty$$

for which the choice of

$$\alpha_k = \frac{A}{B+k} \tag{2}$$

is commonly used, where A and B are constants that must be specified by the user. This specification requires experimentation and care: there can be significant variation in performance depending on choice of these constants. Some recent research show that α_k sequences that go to 0 slower than $1/k$ can result in an improved rate of convergence, where rate of convergence is measured by the asymptotic covariance of the normalized estimates about their limit point (Kushner and Yin, 1997, Chapter 11).

The conditions on U_k are more restrictive. Popular approaches include constraining the sequence of estimators η_k to a compact set specified *a priori*, or assuming that the noise component of U_k be a martingale difference sequence. As commonly observed (Andrieu, Moulines and Priouret, 2005; Chen, 2002; Liang, 2010), these may be difficult to satisfy in practice. See Andrieu, Moulines and Priouret (2005); Liang (2010) for recent developments that impose less restrictive conditions using truncated updates.

An issue for any recursive search algorithm is the choice of starting point. It is often the case that algorithms are good at finding the MLE when the starting point is close to it, but of course the location of the MLE is unknown. For any exponential family with bounded support, Fisher information becomes singular as the natural parameter η goes to ∞ (Rinaldo, Fienberg and Zhou, 2009). Hence methods which rely on the Fisher information matrix may fail when the starting point for η is far from the MLE (Gu and Zhu, 2001; Younes, 1989). Of course, one may try different starting points until a "good" one is found, but this can be cumbersome in practice.

1.2. Algorithm overview

In this article, we propose a simple and practical line search algorithm that converges to the MLE of any regular exponential family when the MLE exists and is unique and the first derivative of the log likelihood can be calculated exactly. When it cannot, the first derivative has a particularly convenient form that is easily estimable with MCMC, making the algorithm still useful in application. We also show how to construct and apply confidence intervals in such a setting to increase the probability of convergence.

The appeal of this algorithm is its ease of use: no trial and error is needed. Experimentation with multiple starting points or tuning parameters is not necessary and no unrealistic *a priori* information about the problem need be specified. It is currently used in the `aster2` contributed R package (Geyer, 2010) as the safeguard for steepest ascent and Newton-Raphson iterations in finding the MLE for aster models.

Our algorithm generates iterated estimates η_k of the MLE $\hat{\eta}$ with the update

$$\eta_{k+1} = \eta_k + \alpha_k p_k \tag{3}$$

where α_k is a *step size* and p_k is a *search direction* and is restricted to be an ascent direction of the log likelihood. Despite the visual similarity between (1) and (3), the line search approach treats the search direction p_k in (3) as constant whereas in SA the corresponding U_k in (1) is random. Furthermore, line search algorithms have more restrictions on the step size α_k. The step size conditions in the classical gradient ascent algorithm, which is the basis for our algorithm, force a sufficiently large increase in the objective function at every step, guaranteeing convergence to the global maximum, if it exists.

Theorem 3.2 in Nocedal and Wright (1999) implies the global convergence of the steepest ascent algorithm for a continuously differentiable function, $\ell(\eta)$. It requires the step length α_k to satisfy the Wolfe conditions for *sufficient increase* and *curvature*:

$$\begin{aligned} \ell(\eta_k + \alpha_k \eta_k) \geq \ell(\eta_k) + c_1 \alpha_k \nabla \ell(\eta_k)^T p_k \\ \nabla \ell(\eta_k + \alpha_k p_k)^T p_k \leq c_2 \nabla \ell(\eta_k)^T p_k \end{aligned} \tag{4}$$

where ∇ is the gradient operator and $0 < c_1 < c_2 < 1$. Variations of these conditions exist in the numerical optimization literature (Nocedal and Wright, 1999; Sun and Yuan, 2006), but all require evaluating the objective function.

Exponential families we consider are an unusual case in optimization in that the objective function is harder to compute than its derivatives and hence not previously considered by optimization theorists. In our algorithm, we replace (4) with a single modified curvature condition:

$$0 \leq \nabla \ell(\eta_k + \alpha_k p_k)^T p_k \leq c \nabla \ell(\eta_k)^T p_k \tag{5}$$

for some $0 < c < 1$. This replacement is possible while still guaranteeing sufficient increase and convergence to the MLE (if it exists) because we have the additional property that the exponential family log likelihood function we consider is strictly concave. The restrictions on the step size α_k along a particular direction p_k and the resulting values for $\ell(\eta_{k+1})$ are depicted in Figure 2.

The desire to avoid calculation of higher order derivatives is motivated not just by computational considerations, but also by how much useful information can be extracted from them. As noted previously, if η is far from the MLE, the Fisher information matrix may be near-singular and algorithms like (unsafeguarded) Newton-Raphson algorithm may fail. For this reason, the best use of our algorithm may be from "long range," filling a gap in the MLE estimation toolbox. It may be expedient to switch to another algorithm like Newton-Raphson after significant progress is made and the Fisher information matrix becomes useful. Our line search algorithm with p_k the Newton direction provides a safeguard for Newton-Raphson that makes it safe (not necessarily efficient) for use from any range. The `aster2` contributed R package (Geyer, 2010) switches p_k from steepest ascent direction to Newton direction after a fixed number of steps ($d/2$ where d is the dimension η) but always finds a step length α_k satisfying (5), iterating until the (unsafeguarded) Newton step satisfies (5).

Our algorithm can be outlined as follows. Let $\|\cdot\|$ denote the Euclidean norm function, and ϵ a small value greater than 0.

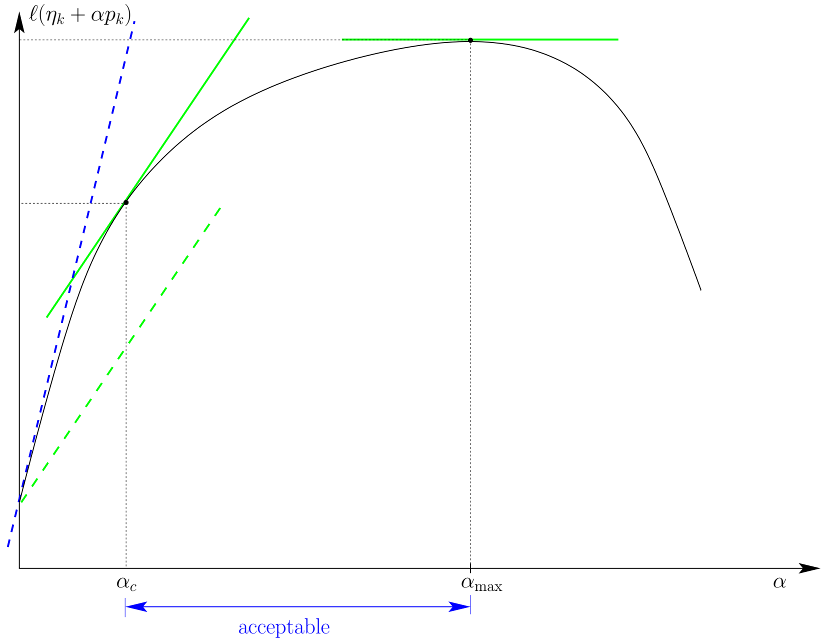

Fig 2. *The acceptable region for step size α_k along a particular search direction p_k according to the modified curvature condition (5). The step sizes α_c and α_{max} correspond to values of $\nabla\ell(\eta_k + \alpha p_k)^T p_k$ equaling $c\nabla\ell(\eta_k)^T p_k$ and 0, respectively. The condition ensures sufficient increase in the log likelihood along the search direction p_k.*

```
Get an initial value, $\eta_0$.
Set $k = 0$.
Calculate $\nabla\ell(\eta_k)$, the direction of steepest ascent.
Set $p_k = \nabla\ell(\eta_k)$.
while $\|\nabla\ell(\eta_k)\| > \epsilon$
    Find a step size $\alpha_k$ that satisfies the modified curvature condition
        $0 \leq \nabla\ell(\eta_k + \alpha_k p_k)^T p_k \leq c\nabla\ell(\eta_k)^T p_k$
    for some $0 < c < 1$.
    $\eta_{k+1} = \eta_k + \alpha_k p_k$.
    Calculate $\nabla\ell(\eta_{k+1})$.
    Find the new search direction $p_{k+1}$, which must be an ascent direction.
    $k = k + 1$.
end while
```

2. Background exponential family theory

An exponential family of distributions (Barndorff-Nielsen, 1978; Geyer, 2009a) on a sample space $\mathcal{Y}$ has log likelihood

$$\ell(\eta) = \langle g(y), \eta \rangle - c(\eta) \tag{6}$$

where $g(y)$ is a d-dimensional vector of *natural statistics*, η a d-dimensional vector of *natural parameters*, and $\langle \cdot\, , \cdot \rangle$ denotes the bilinear form

$$\langle g, \eta \rangle = \sum_{i=1}^{d} g_i \eta_i .$$

So that the probability function integrates to 1, the *cumulant function* c must have the form

$$c(\eta) = \log \left(\int e^{\langle g(y), \eta \rangle} \, d\mu(y) \right), \tag{7}$$

where μ is a measure on $\mathcal{Y}$. Define the *natural parameter space* Ξ as the set of points $\eta = (\eta_1, \ldots, \eta_d)$ that are parameter values indexing distributions in the model. An exponential family is *full* if the natural parameter space is

$$\Xi = \{\eta \in \mathbb{R}^d : c(\eta) < \infty\}, \tag{8}$$

and *regular* if, in addition, Ξ is an open set. We say an exponential family is *minimal* if $g(y)$ is not concentrated on a hyperplane. Minimality guarantees that if an MLE exists, it is unique (Geyer, 2009a).

In finite state space models with complicated dependence like an Ising model or exponential random graph model, (7) is a sum which may have no simple expression and can only be evaluated by explicitly doing the sum. When the sample space $\mathcal{Y}$ is even moderately large, this can be prohibitively expensive. For example, the sample space $\mathcal{Y}$ for an Ising model defined on a 32×32 square lattice where each entry takes values of 0 or 1 has $2^{1024} \approx 10^{300}$ elements. A loop with this many iterations takes too long no matter how programmed.

A useful property of all exponential families (Lehmann and Casella, 1998, p. 27) when η is in the interior of Ξ is that

$$\begin{aligned} \mathrm{E}_\eta(g(Y)) &= \nabla c(\eta) \\ \mathrm{Var}_\eta(g(Y)) &= \nabla^2 c(\eta). \end{aligned}$$

Thus we can express first and second derivatives of the log likelihood (6) and Fisher information, $I(\eta)$, as

$$\begin{aligned} \nabla \ell(\eta) &= g(y) - \mathrm{E}_\eta\, g(Y) && (9) \\ \nabla^2 \ell(\eta) &= -\operatorname{Var}_\eta g(Y) && (10) \\ \mathrm{I}(\eta) &= -\mathrm{E}_\eta \nabla^2 \ell(\eta) = \operatorname{Var}_\eta g(Y) && (11) \end{aligned}$$

and thereby avoid evaluation of the problematic cumulant function c.

3. Long range search algorithm for MLEs

We now present our search algorithm, which will converge to the optimum for any continuously differentiable, strictly convex (or concave) function. The algorithm and requirements are presented in Theorem 3.1. Proofs are in Appendix A.

We apply the algorithm in Theorem 3.2 to the specific setting of finding the MLE in a regular exponential family when the MLE is known to exist and be unique and the gradient can be calculated exactly.

In order to be consistent with the general optimization literature (Fletcher, 1987; Nocedal and Wright, 1999), we state our algorithm in this section from the perspective of a minimization problem. Thus we wish to minimize a real-valued objective function f defined on $\mathbb{R}^n$.

Theorem 3.1 (Convex function root search). *Consider any line search of the form*

$$x_{k+1} = x_k + \alpha_k p_k \tag{12}$$

used to minimize the objective function f, which satisfies the following assumptions:

1. *The objective function f is bounded below in $\mathbb{R}^n$.*
2. *The objective function f is proper, lower semicontinuous, and strictly convex.*
3. *The objective function f is differentiable in an open set $\mathcal{N}$ containing the level set $\operatorname{lev}_{\leq f(x_0)} f$, which is bounded, where x_0 is the starting point of the iteration.*

4. *The* search direction p_k *is a non-zero* descent direction *such that the angle θ_k between the search direction p_k and steepest descent direction $-\nabla f(x_k)$ is restricted to be less than 90 degrees by*

$$\cos \theta_k \geq \delta > 0$$

for some fixed $\delta > 0$.

Then, unless $\nabla f(x_k) = 0$, in which case x_k is already the solution and the search is complete, it is possible to find a step length α_k that satisfies the curvature condition

$$c \nabla f(x_k)^T p_k \leq \nabla f(x_k + \alpha_k p_k)^T p_k \leq 0 \tag{13}$$

for some fixed $0 < c < 1$.

Furthermore, repeated iterations of (12) *satisfying assumptions 1 through 4 and* (13) *will produce a sequence, $x_1, x_2, \ldots$ such that*

$$\lim_{k \to \infty} ||\nabla f(x_k)|| = 0.$$

We apply Theorem 3.1 to the setting of exponential families to find the MLE when it exists.

Theorem 3.2. *For a regular exponential family with minimal representation where the MLE exists, the line search described in Theorem 3.1 can be applied to the negative log likelihood function $-\ell(\eta)$ so that a search starting at any $\eta_0 \in \Xi$ will converge to the MLE of η.*

The issue of MLE existence is a problem in computational geometry, not an optimization problem, so we do not address it here. See Geyer (2009a); Okabayashi (2011); Rinaldo, Fienberg and Zhou (2009).

4. Refinements of algorithm

In Theorem 3.1, we restricted our search direction p_k to be a descent direction, so that $\nabla f(x_k)^T p_k < 0$ or, alternatively, the angle θ_k between the search direction p_k and steepest descent direction $-\nabla f(x_k)$ is less than 90 degrees. However, this still leaves many possibilities for the choice of p_k other than steepest descent. In addition, we have specified restrictions on the step size α_k in the curvature condition (13) with $0 < c < 1$, but it would be useful to know if certain values of c are better than others.

4.1. Search directions

In our examples in Section 5, we default to steepest descent directions in our implementation for transparency. Although often effective in early steps, steepest descent directions can result in a zigzagging trajectory of the sequence x_k

(Sun and Yuan, 2006, Section 3.1). Conjugate gradient methods address this phenomena and cover the sample space more efficiently (Nocedal and Wright, 1999, Chapter 5). It is easy to implement a variant of the Polak-Ribière method (Nocedal and Wright, 1999, pp. 120–122) here, requiring little more in terms of calculation or storage. The search direction p_k would update with an extra intermediate step as follows:

$$\begin{aligned}\gamma_{k+1}^{PR} &= \max\left(0, \frac{[\nabla f(x_{k+1})]^T(\nabla f(x_{k+1}) - \nabla f(x_k))}{\|\nabla f(x_k)\|^2}\right) \\ p_{k+1} &= -\nabla f(x_{k+1}) + \gamma_{k+1}^{PR}\, p_k.\end{aligned}$$

Note that when $\gamma_{k+1}^{PR} = 0$, p_{k+1} will be just $-\nabla f(x_{k+1})$, the direction of steepest descent, and thus serves as a "reset". The curvature condition (13) guarantees that this method always yields a descent direction for p_{k+1} and thus Theorem 3.1 still holds.

4.2. Step size

We now turn our attention to the optimal step size α_k when our objective function is the log likelihood of an exponential family. Taking the derivative of $\ell(\eta_k + \alpha_k p_k)$ with respect to α_k shows that the log likelihood is maximized as a function of α_k along the direction p_k when

$$\nabla \ell(\eta_{k+1})^T p_k = 0.$$

By choosing c to be small, say 0.2, we ensure that the step taken is close to maximizing the log likelihood along the search direction. This is also apparent in Figure 2.

Making c too small, however, may make it difficult to find an α_k that meets the curvature condition (5) since this search must be done numerically. In fact, as the line search nears the MLE and $\nabla \ell(\eta_k)$ gets smaller, the rightmost term in (5) gets smaller in magnitude (it equals $c\|\nabla \ell(\eta_k)\|^2$ if using steepest ascent directions), making a numerical search for α_k more challenging.

4.3. MCMC approximations

Our algorithm requires us to be able to calculate $\nabla \ell(\eta)$ using (9). For many applications, we will need to approximate $\mathrm{E}_\eta\, g(Y)$ using MCMC. That is,

$$\nabla \ell(\eta) = g(y) - \mathrm{E}_\eta\, g(Y) \approx g(y) - \frac{1}{m}\sum_{i=1}^{m} g(Y_i), \tag{14}$$

where $Y_1, \ldots, Y_m$ are MCMC draws from the distribution with parameter η. There are many MCMC algorithms such as Metropolis-Hastings (Geyer, 2011) or Swensen-Wang (Swendsen and Wang, 1987), used for the Ising model example in Section 5.2. We show examples in the next section where $\nabla\ell(\eta)$ can be calculated exactly and where it must be approximated.

The accuracy of the approximation in (14) increases with Monte Carlo sample size m. When the current estimate is far away from the MLE, we can use smaller m to save time and work with a fairly noisy approximation of the gradient. However, when the current estimate approaches the MLE, larger m are necessary.

Our algorithm relies on the computed values of $\nabla\ell(\eta)$ in the curvature condition (5), as well as the stop condition for the algorithm, $\|\nabla\ell(\eta_k)\| < \epsilon$. Given that we may only have approximations of $\nabla\ell(\eta)$, we cannot know for certain if either of these conditions are truly met. We can ameliorate this by constructing confidence intervals for each of the inequalities.

For the inequalities in (5), we can estimate asymptotic standard errors of $\nabla\ell(\eta_k + \alpha_k p_k)^T p_k$ and $c\nabla\ell(\eta_k)^T p_k - \nabla\ell(\eta_k + \alpha_k p_k)^T p_k$ by appealing to the Markov chain Central limit theorem (Chan and Geyer, 1994; Jones, 2004; Roberts and Rosenthal, 1997, 2004). The `initseq` function from the R package `mcmc` (Geyer, 2009b) can be used to estimate asymptotic standard errors for univariate functionals of reversible Markov chains: given an MCMC sample for a univariate quantity, `initseq` returns a value (divided by sample size) that is an estimate of the asymptotic variance in the Markov chain central limit theorem. Both of the quantities in (5) are univariate. In the second expression, $c\nabla\ell(\eta_k)^T p_k - \nabla\ell(\eta_k + \alpha_k p_k)^T p_k$, the MCMC sample generated for $\nabla\ell(\eta_k + \alpha_k p_k)^T p_k$ is independent of the sample generated for $c\nabla\ell(\eta_k)^T p_k$. Thus `initseq` can be applied to each sample separately and the results summed for an estimated variance. We can then be approximately 95% confident (non-simultaneously) that α_k satisfies (5) if

$$\begin{aligned} \nabla\ell(\eta_k + \alpha_k p_k)^T p_k - 1.645 \cdot \mathrm{se}_1 &> 0 \\ c\nabla\ell(\eta_k)^T p_k - \nabla\ell(\eta_k + \alpha_k p_k)^T p_k - 1.645 \cdot \mathrm{se}_2 &> 0 \end{aligned}$$

where se_1 and se_2 are the asymptotic standard errors for $\nabla\ell(\eta_k + \alpha_k p_k)^T p_k$ and $c\nabla\ell(\eta_k)^T p_k - \nabla\ell(\eta_k + \alpha_k p_k)^T p_k$, respectively, calculated as described.

The delta method can be applied to estimate a standard error for $\|\nabla\ell(\eta_k)\|$. The asymptotic variance is calculated by

$$V\left(\|\nabla\ell(\eta_k)\|\right) = \frac{1}{\|\nabla\ell(\eta_k)\|^2} \nabla\ell(\eta_k)^T \,\Sigma\, \nabla\ell(\eta_k),$$

where Σ is the variance matrix of $\nabla\ell(\eta_k)$ and can be estimated by the sample variance matrix of the batch mean vectors of $g(Y_1), \ldots, g(Y_n)$ divided by the number of batches (the `initseq` function requires a univariate vector and so cannot be used here). We can be approximately 95% confident that $\|\nabla\ell(\eta_k)\| > \epsilon$ if

$$\|\nabla\ell(\eta_k)\| - 1.645\sqrt{V\left(\|\nabla\ell(\eta_k)\|\right)} > \epsilon.$$

In practice, however, use of confidence intervals does not appear necessary with Monte Carlo sample sizes that are set large enough so that these standard errors are initially small relative to the point estimates. The ratio of point estimate to standard error of course decreases as the algorithm progresses and the estimate of the parameter nears the MLE, reflected in $\nabla \ell(\eta_k)$ nearing 0. Thus these confidence intervals are most useful as a guide for when to increase the MCMC sample size, or when to switch methods, or when to terminate the algorithm.

4.4. Combining with other algorithms

We believe the best use of this algorithm is in combination with other faster methods like MCMC-MLE (Geyer and Thompson, 1992) or Newton-Raphson safeguarded by our line search algorithm. Our algorithm with steepest ascent or conjugate gradient search direction should be used initially from "long range", when one has no good intuition for an initial value. It is well known that when the objective function is quadratic, the conjugate gradient method with exact arithmetic converges to the solution in at most d steps, where d is the dimension of the problem (Nocedal and Wright, 1999, Chapter 5). As a rule of thumb, we think using our algorithm for $2d$ steps before switching seems reasonable when using conjugate gradient directions. Determining a more precise criteria for when we are inside the "radius of convergence" for algorithms like Newton-Raphson or MCMC-MLE is an area for further research.

5. Examples

5.1. Example: logistic regression

We illustrate the application of our algorithm in the case of a logistic regression with a starting point far from the solution. In such a case, the Hessian matrix is often near-singular and algorithms such as Newton-Raphson which rely on it will fail. For classical SA with step size $1/k$, the magnitudes of the updates diminishes too quickly for the parameter estimates to approach the MLE in a reasonable amount of time.

The response vector Y has components that are Bernoulli trials with mean vector p. The natural parameter is $\theta_i = \log\left(\frac{p_i}{1-p_i}\right)$, which is modeled componentwise as a linear function of the predictors $1, x_1, \ldots, x_q$, so that

$$\theta_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_q x_{q\,i} = \beta^T x_i \qquad i = 1, \ldots, n$$

where $\beta = (\beta_0, \ldots, \beta_q)^T$ and $x_i = (1, x_{1i}, \ldots, x_{qi})^T$.

Defining the model matrix M to be the $n \times (q+1)$ matrix with the x_i as rows, we can express $\theta = M\beta$. This in turn allows us to reparameterize the exponential family as one with β as the natural parameter vector and $M^T y$ the vector of statistics with log likelihood

$$\ell(\beta) = \beta^T(M^T y) - c(\beta),$$

where y is the vector of observed Bernoulli responses. By (9), the gradient is

$$\nabla \ell(\beta) = M^T y - \mathrm{E}_\beta(M^T Y) = M^T(y - \mathrm{E}_\beta(Y)),$$

where $\mathrm{E}_\beta(Y) = \frac{1}{1+\exp(-M\beta)}$ can be calculated exactly. This allows us to directly apply Theorem 3.2.

Suppose we specify our true parameter value to be $\beta = (0, 2, 2, 1, 1, 0, 0, 0)^T$ and use 100 independent draws from a correlated multivariate normal distribution centered at 0 as our predictors to generate 100 independent Bernoulli trials. Fitting these data using the R function `glm`, we find the MLE of β to be

$$\hat{\beta}_{\mathrm{MLE}} = (0.635, 5.949, 1.273, 0.180, 1.006, 1.536, -2.252, -0.472)^T,$$

where the disparity to the true value of β results from a relatively small sample size of $n = 100$. We use $\beta_0 = (5, 4, 3, 2, 1, 0, -1, -2)^T$ for the starting point for our line search algorithm, a point for which Newton-Raphson fails due to a nearly singular Hessian matrix.

We measure the performance of our algorithm in terms of the total number of iterations used, where each iteration requires evaluation of the gradient, $\nabla\ell(\beta_k + \alpha_k p_k)$. Typically, several iterations take place in an inner loop to find a step size α_k that meets the curvature condition (5), a process that grows increasingly difficult as the estimates near the MLE since the rightmost term in (5) gets smaller in magnitude. Once an acceptable step size is found, the parameter estimate β_k is updated and a new search direction is determined, requiring another evaluation of the gradient.

Our algorithm took 54 iterations over 20 different search directions to get $\|\nabla\ell(\beta_k)\| < 0.01$ and arrive at an estimate for the MLE that differs from the `glm` result by 0.0117 in Euclidean distance (See Table 1). Using the Polak-Ribière conjugate gradient method described in the previous section resulted in comparably sharp MLE estimates (see Table 1) in fewer iterations—28 over 11 search directions—a noticeable improvement.

We also applied SA with step size $1/k$ (setting $A = 1$, $B = 0$ in (2)) from the same starting point β_0. The choice of constants A and B in the step size is of course not likely to be optimal; however, we want to apply SA without trial and error experimentation. After 10,000 iterations, the parameter estimates look nothing at all like the MLE (See Table 1). The starting point β_0 is so far from the MLE and the step sizes so small that the algorithm does not converge in a reasonable amount of time. Table 2 shows the first 20 step sizes used by SA and our line search. Our line search continues to use step sizes of relatively large magnitude even well into the process. It should be noted that these 20 step sizes correspond to the first 20 iterations of SA but all 54 iterations of our line search algorithm since it spends several iterations finding an acceptable step size for each update.

Table 1

Comparison of MLEs of β for Example 1: MLE = `glm`, Steep = line search using steepest ascent, CG = line search using conjugate gradient, and SA = SA with step size = $1/k$ terminated at 10,000 iterations, n = number of iterations. Our proposed algorithm arrives at nearly identical MLE estimates to `glm`.

	n	$\beta[1]$	$\beta[2]$	$\beta[3]$	$\beta[4]$	$\beta[5]$	$\beta[6]$	$\beta[7]$	$\beta[8]$
True β		0.000	2.000	2.000	1.000	1.000	0.000	0.000	0.000
$\hat{\beta}_{\text{MLE}}$		0.635	5.949	1.273	0.180	1.006	1.536	-2.252	-0.472
$\hat{\beta}_{\text{Steep}}$	54	0.633	5.938	1.272	0.181	1.005	1.535	-2.249	-0.470
$\hat{\beta}_{\text{CG}}$	28	0.631	5.936	1.272	0.181	1.003	1.532	-2.244	-0.470
$\hat{\beta}_{\text{SA}}$	10^4	1.280	10.619	5.588	4.005	2.478	-7.153	1.255	0.264

Table 2

The first 20 step sizes used by SA (with step size $1/k$) and our algorithm for Example 1. The step sizes used by our algorithm do not diminish like $1/k$.

k	$\alpha_{\text{SA}} = 1/k$	α_{Steep}	α_{CG}
1	1.000	0.192	0.192
2	0.500	0.319	0.319
3	0.333	0.403	0.416
4	0.250	0.353	0.561
5	0.200	0.380	0.491
6	0.167	0.333	1.092
7	0.143	0.420	0.359
8	0.125	0.307	0.314
9	0.111	0.442	0.275
10	0.100	0.283	0.318
11	0.091	0.483	0.278
12	0.083	0.241	-
13	0.077	0.745	-
14	0.071	0.203	-
15	0.067	1.224	-
16	0.063	0.173	-
17	0.059	2.510	-
18	0.056	0.195	-
19	0.053	0.944	-
20	0.050	0.173	-

5.2. Example: Ising model

In this example, we apply our gradient-based line search algorithm to an Ising model (Ising, 1925) on a toroidal square lattice. Ising models are exponential families where each entry in the square lattice takes the value of either zero or one. A realized sample is shown in Figure 1. The sufficient statistic vector is two-dimensional, comprising the number of entries with value one and the number of "neighbor" entries with the same value. Entries are considered "neighbors" if they are adjacent to one another horizontally or vertically (but not diagonally).

Here we describe the toroidal square lattice as an $n \times n$ matrix Y and each entry as Y_{ij}, where i and j take values in $1, \ldots, n$ considered as a cyclical set (addition is done modulo n). The sufficient statistic, $g(y)$, has components:

$$\begin{aligned} g_1(y) &= \sum_{i=1}^{n} \sum_{j=1}^{n} I(Y_{ij} = 1), \\ g_2(y) &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \big[I(Y_{ij} = Y_{i-1,j}) + I(Y_{ij} = Y_{i,j-1}) \\ &\qquad\qquad + I(Y_{ij} = Y_{i+1,j}) + I(Y_{ij} = Y_{i,j+1}) \big], \end{aligned}$$

where $I(\cdot)$ denotes the indicator function taking logical expressions to the numbers zero and one, false expressions to zero and true expressions to one.

Because of the interdependence of neighboring entries in the lattice, there is no closed form expressing $\nabla \ell(\eta)$ as in the logistic example. Instead, we need to approximate $\nabla \ell(\eta)$ using MCMC as described by (14). As discussed in Section 4.3, Theorem 3.2 cannot be applied directly, but as we demonstrate here, satisfactory estimates are still attained. The MCMC draws are performed here using the Swendsen-Wang algorithm (Swendsen and Wang, 1987; Wang and Swendsen, 1990), available in the contributed R package `potts` (Geyer and Johnson, 2010).

We choose $\eta = \big(0, \log(1 + \sqrt{2})\big)^T$ to generate a 32×32 lattice, which we use as our observed data (Figure 1). This value for η is of particular interest because it corresponds to the phase transition point (Potts, 1952) and has been shown to be difficult to estimate (Geyer, 1990). In order to get a good estimate of the MLE to which we can compare our algorithm's results, we use 10 iterations of MCMC Newton-Raphson (Penttinen, 1984) starting at the true value of η so that it will converge.

We apply our line search algorithm to this data using a far off initial value of $\eta^{(0)} = (2, 0.001)$ and a fixed MCMC sample size of 10,000. Our algorithm used 62 iterations (gradient evaluations) over 17 search directions to get $\|\nabla \ell(\eta_k)\| < 0.005$ and arrive at an estimate of the MLE that differs from Newton-Raphson by 0.0037 (see Table 3). Using the Polak-Ribière conjugate gradient method resulted in comparably sharp MLE estimates using 45 iterations over 7 search directions. The total MCMC sample sizes used were $62 \times 10,0000 = 620,000$ and $45 \times 10,0000 = 450,000$, respectively.

Table 3

Comparison of MLEs for η for Example 2: MLE = Newton-Raphson starting from the true η, Steep = line search using steepest ascent, CG = line search using conjugate gradient, and SA = SA with step size = $1/k$. All algorithms converged.

	MC Samples (thousands)	$\eta[1]$	$\eta[2]$
True η		0.000	0.881
$\hat{\eta}_{\text{MLE}}$		-0.007	0.896
$\hat{\eta}_{\text{Steep}}$	620	-0.011	0.895
$\hat{\eta}_{\text{CG}}$	450	-0.008	0.895
$\hat{\eta}_{\text{SA}}$	1368	-0.010	0.895

Table 4

The first 17 step sizes used by SA (with step size $1/k$) and our algorithm for Example 2. The step sizes used by our algorithm are initially much smaller than $1/k$.

k	$\alpha_{\text{SA}} = 1/k$	α_{Steep}	α_{CG}
1	1.0000	0.0029	0.0029
2	0.5000	0.0005	0.0005
3	0.3333	0.0017	0.0017
4	0.2500	0.0013	0.0045
5	0.2000	0.0017	0.0007
6	0.1667	0.0011	0.0002
7	0.1429	0.0021	0.0015
8	0.1250	0.0009	
9	0.1111	0.0020	
10	0.1000	0.0007	
11	0.0909	0.0018	
12	0.0833	0.0006	
13	0.0769	0.0013	
14	0.0714	0.0006	
15	0.0667	0.0007	
16	0.0625	0.0003	
17	0.0588	0.0013	

We also applied MCMC SA, again with step size $1/k$ from the same starting point $\eta^{(0)}$, and used a MCMC sample size of 1,000 for gradient calculation. Here SA converged in 1368 iterations or 1,368,000 MC samples, comparable to our algorithm (see Table 3). Table 4 shows the first 17 step sizes used by SA and our line search. The step sizes used by our line search are initially very small compared to $1/k$, but stay in a range of about 1/300 to 1/3000. So, the $1/k$ step size used by SA in fact occasionally satisfies our curvature condition when k is large.

6. Discussion

We have presented a simple line search algorithm for finding the MLE of a regular exponential family when the MLE exists. The algorithm avoids the trial and error experimentation of tuning parameters and starting points commonly associated with optimization routines not invented by optimization specialists.

Our algorithm is modeled after algorithms discussed in optimization textbooks (Fletcher, 1987; Nocedal and Wright, 1999; Sun and Yuan, 2006), all of which are safeguarded to ensure rapid automatic convergence.

Convergence is guaranteed when the gradient can be calculated exactly. Even when the gradient cannot be calculated exactly and is only estimable via MCMC, the algorithm is still useful in practice, as demonstrated by the Ising model example. We have also described a way to construct and use confidence intervals to make convergence highly probable.

The algorithm can be computationally demanding. When the current iteration approaches the solution, the curvature condition for step size becomes more difficult to satisfy and the method may require several iterations of MCMC sampling and perhaps an increase in MCMC sample size. Eventual increase in MCMC sample size is unavoidable, because the achievable accuracy is inversely proportional to the square root of the MCMC sample size, as in all Monte Carlo. Thus we believe the best use of this algorithm is in combination with other faster methods like MCMC-MLE (Geyer and Thompson, 1992) or Newton-Raphson safeguarded by our line search algorithm. Our algorithm should be used from "long range", when one has no good intuition for an initial value and is concerned about picking one that is far from the MLE. The switch between types of search direction (steepest ascent, conjugate gradient, or Newton) within our algorithm or the switch to another algorithm (such as MCMC-MLE (Geyer and Thompson, 1992)) need not require manual intervention. When used in combination, we do not think the confidence intervals are necessary as the curvature condition is quite easily satisfied when the current iteration is far from the MLE.

One way to improve performance is to use conjugate gradient search directions rather than steepest ascent. In our examples, this reduced the number of iterations by over 25%. However, in other problems we tried with different dimensionality, this performance varied significantly and it appears that no guarantee can be made about quantity of improvement in performance, though in all cases we examined, it never did worse. This is no surprise, because the necessity of "preconditioning" for good performance of the conjugate gradient algorithm is well known (but no good "preconditioner" is available for maximum likelihood in exponential families).

There are several outstanding issues. Most notably, we have not showed convergence of the algorithm when the gradient is approximated via MCMC. This is a more difficult theoretical problem and is the motivation for stochastic approximation research. Further work is necessary to determine if one can adapt our restrictive curvature condition (13) to the approach of Andrieu, Moulines and Priouret (2005); Liang (2010) in MCMC stochastic approximation.

Another remaining issue is the stopping criteria: what value should be chosen for ϵ in the exit condition $\|\nabla\ell(\eta_k)\| < \epsilon$? Because the value of $\|\nabla\ell(\eta_k)\|$ can only be approximated via MCMC, one cannot be certain if this condition is actually satisfied. Here again, the switch to another methodology may be appropriate, though at least in our Ising model example, our use of 10,000 for the MCMC sample size and 0.005 for ϵ were successful in obtaining a reasonable parameter estimate.

A final remaining issue is estimation of Monte Carlo error of the estimates. Here too we recommend switching to another algorithm at the end. The MCMC-MLE procedure gives accurate error estimates (Geyer, 1994). For very small steps these are essentially the same as the Monte Carlo error of a single unsafeguarded Newton-Raphson step, so the method in Geyer (1994) can be used for either.

Appendix A: Proofs

Our algorithm minimizes the objective function f by performing repeated one-dimensional updates. We need the following lemma to transfer global properties of the objective function to the objective function restricted to a search direction.

Lemma A.1. *Suppose a function $f : \mathbb{R}^n \to \mathbb{R}$ is proper, lower semicontinuous, and strictly convex. Then the minimum for f exists and is unique if and only if every nonempty level set $\operatorname{lev}_{\leq \alpha} f$ is bounded.*

Proof of Lemma A.1. Assume every non-empty level set is bounded. Then by Theorem 1.9 in Rockafellar and Wets (2004), the minimum of f is finite and thus exists. By strict convexity, this minimum is also unique.

Now assume the minimum for f exists, denoted by $\min f$. By assumption, the level set $\operatorname{lev}_{\leq \min f} f$ contains exactly one point. By Corollary 8.7.1 in Rockafellar (1970), the level sets $\operatorname{lev}_{\leq \alpha} f$ are bounded for every α. □

Proof of Theorem 3.1. The objective function f is bounded below, strictly convex, lower semicontinuous, and the level set $\operatorname{lev}_{\leq f(x_0)} f$ is bounded by assumption, where x_0 is the starting point of the iteration. By Theorem 1.9 in Rockafellar and Wets (2004), the global minimum exists. Then by Lemma A.1, all level sets of type $\operatorname{lev}_{\leq a} f$, $a \in \mathbb{R}$ are bounded in $\mathbb{R}^n$. Restricting the set to be along a search direction p_k maintains the boundedness of these sets. By Lemma A.1 again, the minimum in this restriction exists and is unique.

Then, unless $\nabla f(x_k) = 0$ in which case x_k is already the solution, for each k, we can uniquely define α_{c_k} and α_{min_k} as follows:

$$\nabla f(x_k + \alpha_{c_k} p_k)^T p_k = c \nabla f(x_k)^T p_k \tag{15}$$

$$\nabla f(x_k + \alpha_{min_k} p_k)^T p_k = 0. \tag{16}$$

The point α_{c_k} is uniquely defined because it is the minimizer of $\alpha \mapsto f(x_k + \alpha p_k) - \alpha c \nabla f(x_k)^T p_k$. These values appear on the α-axis in Figure 3.

By the strict convexity of f and Theorem 2.14 in Rockafellar and Wets (2004),

$$f(x_k + \alpha_{c_k} p_k) < f(x_k) + \left[\nabla f(x_k + \alpha_{c_k} p_k)\right]^T \alpha_{c_k} p_k.$$

Applying (15) to the right hand side of the above gives

$$f(x_k + \alpha_{c_k} p_k) < f(x_k) + \alpha_{c_k} c \nabla f(x_k)^T p_k. \tag{17}$$

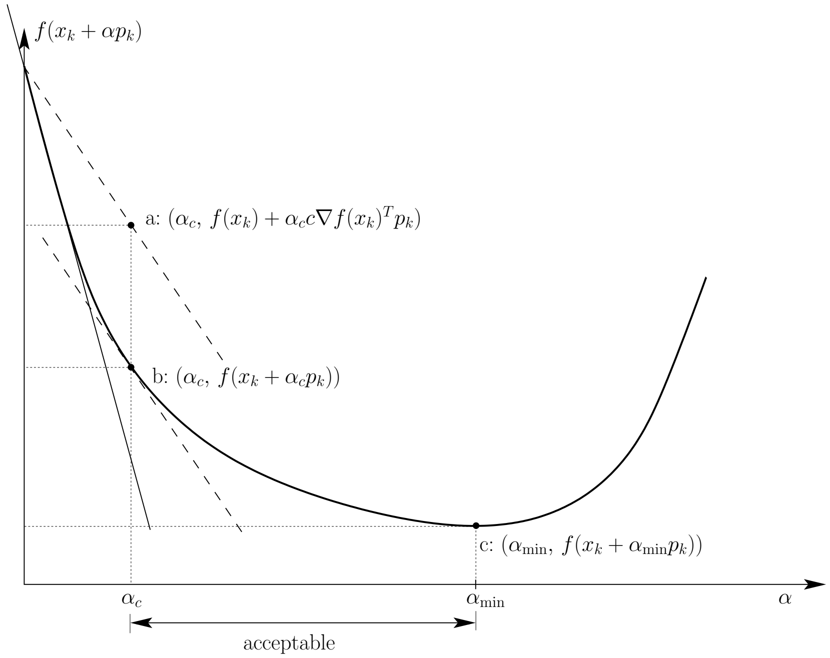
$f(x_k + \alpha p_k)$
a: $(\alpha_c,\ f(x_k) + \alpha_c c \nabla f(x_k)^T p_k)$
b: $(\alpha_c,\ f(x_k + \alpha_c p_k))$
c: $(\alpha_{\min},\ f(x_k + \alpha_{\min} p_k))$
α_c
$\alpha_{\min}$
α
acceptable

Fig 3. *The acceptable region for α according to the curvature condition* (13) *when restricting f to direction p_k.*

(See points a and b in Figure 3).

The subproblem $\alpha \mapsto f(x_k + \alpha p_k)$ is strictly convex and hence monotonically decreasing at α_k such that $\alpha_{c_k} \leq \alpha_k \leq \alpha_{min_k}$ (in Figure 3, see points b and c). That is,

$$f(x_k + \alpha_{min} p_k) \leq f(x_k + \alpha_k p_k) \leq f(x_k + \alpha_{c_k} p_k). \tag{18}$$

Combining the second inequality of (18) with (17), we have

$$f(x_k + \alpha_k p_k) < f(x_k) + \alpha_{c_k} c \nabla f(x_k)^T p_k, \tag{19}$$

which can be rearranged as

$$f(x_k) - f(x_k + \alpha_k p_k) > -\alpha_{c_k} c \nabla f(x_k)^T p_k. \tag{20}$$

This last inequality (20) expresses a lower bound for the amount of decrease in our objective function at each step (the right-hand side is positive since $\nabla f(x_k)^T p_k < 0$ by assumption that p_k is a descent direction). It is this lower bound that we will use to cover the distance to the minimum of the objective function.

We now turn our attention to (15). Define $x_{c_k} = x_k + \alpha_{c_k} p_k$. Then

$$\nabla f(x_{c_k})^T p_k = c \nabla f(x_k)^T p_k.$$

Subtracting $\nabla f(x_k)^T p_k$ from both sides gives

$$\left(\nabla f(x_{c_k}) - \nabla f(x_k)\right)^T p_k = (c-1) \nabla f(x_k)^T p_k. \tag{21}$$

By Corollary 25.5.1 in Rockafellar (1970), since f is convex and differentiable on the open convex set $\mathcal{N}$, it is actually continuously differentiable on $\mathcal{N}$. It is then Lipschitz continuously differentiable relative to any compact subset of $\mathcal{N}$.

Applying this to the compact level set $\text{lev}_{\leq f(x_k)} f$, which is contained in $\mathcal{N}$ by assumption, there exists a constant $L < \infty$ such that

$$||\nabla f(x) - \nabla f(\tilde{x})|| \leq L||x - \tilde{x}|| \quad \text{for all } x, \tilde{x} \in \text{lev}_{\leq f(x_0)} f. \tag{22}$$

Applying (22) to x_{c_k} and x_k, we have

$$||\nabla f(x_{c_k}) - \nabla f(x_k)|| \leq L||x_{c_k} - x_k||$$

or

$$||\nabla f(x_{c_k}) - \nabla f(x_k)|| \leq L||\alpha_{c_k} p_k||.$$

Multiplying both sides by $\|p_k\|$ gives

$$||\nabla f(x_{c_k}) - \nabla f(x_k)|| \cdot ||p_k|| \leq \alpha_{c_k} L||p_k||^2$$

and by Cauchy-Schwarz this implies

$$\begin{aligned}(\nabla f(x_{c_k}) - \nabla f(x_k))^T p_k &\leq ||\nabla f(x_{c_k}) - \nabla f(x_k)|| \cdot ||p_k|| \\ &\leq \alpha_{c_k} L ||p_k||^2.\end{aligned} \tag{23}$$

Substituting (21) into the left-hand side of this last inequality (23) gives

$$(c-1)\nabla f(x_k)^T p_k \leq \alpha_{c_k} L ||p_k||^2$$

or

$$-\alpha_{c_k} \leq \frac{(1-c)}{L} \frac{\nabla f(x_k)^T p_k}{||p_k||^2}. \tag{24}$$

Write out the first $k+1$ inequalities of (20):

$$\begin{aligned} f(x_1) &< f(x_0) + \alpha_{c_0} c \nabla f(x_0)^T p_0 \\ f(x_2) &< f(x_1) + \alpha_{c_1} c \nabla f(x_1)^T p_1 \\ &\cdots \\ f(x_k) &< f(x_{k-1}) + \alpha_{c_{k-1}} c \nabla f(x_{k-1})^T p_{k-1} \\ f(x_{k+1}) &< f(x_k) + \alpha_{c_k} c \nabla f(x_k)^T p_k \end{aligned} \tag{25}$$

Telescoping the right-hand side of (25),

$$f(x_{k+1}) < f(x_0) + c \sum_{j=0}^{k} \alpha_{c_j} \nabla f(x_j)^T p_j.$$

Noting that $\nabla f(x_j)^T p_j < 0$, we can substitute our upper bound (24) for $-\alpha_{c_j}$ in the right-hand side above,

$$f(x_{k+1}) < f(x_0) - c \sum_{j=0}^{k} \frac{(1-c)}{L} \frac{\nabla f(x_j)^T p_j}{||p_j||^2} \nabla f(x_j)^T p_j$$

which simplies to

$$f(x_{k+1}) < f(x_0) - c \sum_{j=0}^{k} \frac{(1-c)}{L} \frac{(\nabla f(x_j)^T p_j)^2}{||p_j||^2}.$$

Because $f(x)$ is bounded below by assumption, there exists some $M < \infty$ such that $f(x_0) - f(x_{k+1}) < M$ for all k. Then rearranging the above yields,

$$\frac{c(1-c)}{L} \sum_{j=0}^{k} \frac{(\nabla f(x_j)^T p_j)^2}{||p_j||^2} < M < \infty.$$

The angle θ_j between the search direction p_k and steepest descent direction $-\nabla f_k$ can be expressed by $\cos\theta_j = \frac{-\nabla f(x_j)^T p_j}{||\nabla f_j||\cdot||p_j||}$. Substituting this into the equation above and taking $k \to \infty$,

$$\frac{c(1-c)}{L}\sum_{j=0}^{\infty} \cos^2\theta_j ||\nabla f(x_j)||^2 < \infty.$$

Since $0 < c < 1$,

$$\sum_{j=0}^{\infty} \cos^2\theta_j ||\nabla f(x_j)||^2 < \infty. \tag{26}$$

The convergent series in (26) implies that

$$\cos^2\theta_k ||\nabla f(x_k)||^2 \to 0 \text{ as } k \to \infty.$$

With the additional restriction on the search direction p_k such that $\cos\theta_k \geq \delta > 0$ for some choice of δ, for all choices of k, we get the desired convergence result of

$$\lim_{k\to\infty} ||\nabla f(x_k)|| = 0.$$

□

The inequality (26) has been referred to as *Zoutendijk's condition* (Nocedal and Wright, 1999), though we arrive at this result via different assumptions.

Theorem 3.1 shows that the gradient of the objective function converges to 0. The proof for Theorem 3.2 is concerned with the conditions for mapping this convergence to the convergence of the iterated parameter estimates η_k to the unique MLE. In particular, the mapping from η_k to the gradient must be globally invertible.

Proof of Theorem 3.2. The Fisher information for a regular exponential family is non-singular by (11) and thus invertible. If we consider the map defined by

$$h(\eta) = \nabla c(\eta)$$

where c is the cumulant function (7), its first derivative matrix is

$$\nabla h(\eta) = \nabla^2 c(\eta) = I(\eta) \tag{27}$$

which is again non-singular. Since this is true for any η, by the inverse function theorem, h is everywhere locally invertible.

In fact, h is globally invertible. For any μ in the range of h, consider the function

$$q(\eta) = \mu^T\eta - c(\eta).$$

Since $\nabla^2 q(\eta) = -I(\eta)$ by (27), q is strictly concave. Therefore, a maximizer for q, call it $\hat{\eta}$, is unique if it exists and satisfies the first-order condition

$$\nabla q(\hat{\eta}) = 0.$$

This in turn implies that

$$\mu - h(\hat{\eta}) = 0$$

or

$$\mu = h(\hat{\eta}).$$

Because of the assumption that μ is in the range of h, this means that $\hat{\eta}$ in fact exists, and by the strict concavity of q, is unique. This implies that h must be one-to-one and hence globally invertible.

Since c is infinitely differentiable by Theorem 2.7.1 in Lehmann and Romano (2005), so is h, and by the inverse function theorem, so is h^{-1} (even if we do not know the form of h^{-1}). The first derivative of h^{-1} can be expressed as

$$\nabla h^{-1}(\mu) = [\nabla h(\eta)]^{-1} = [I(\eta)]^{-1}$$

when $\mu = h(\eta)$ and is thus non-singular everywhere, including at the MLE of η, $\hat{\eta}_{\text{MLE}}$.

Thus our algorithm, which concludes that $||\nabla \ell(\eta_k)|| = ||g(y) - h(\eta_k)|| \to 0$, implies that

$$\mu_k = h(\eta_k) \to g(y),$$

or

$$h^{-1}(\mu_k) \to h^{-1}\left(g(y)\right),$$

or

$$\eta_k \to \hat{\eta}_{\text{MLE}}.$$

□

References

Andrieu, C., Moulines, E. and Priouret, P. (2005). Stability of Stochastic Approximation under Verifable Conditions. *SIAM Journal on Control and Optimization* **44** 283–312.

Barndorff-Nielsen, O. (1978). *Information and Exponential Families in Statistical Theory*. John Wiley & Sons.

Besag, J. (1974). Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society,* Series B **36** 192-236.

Besag, J. (1975). Statistical Analysis of Non-lattice Data. *The Statistician* **24** 179-195.

Brown, L. D. (1986). *Fundamentals of Statistical Exponential Families: with Applications in Statistical Decision Theory.* Institute of Mathematical Statistics, Hayward, CA.

Chan, K. S. and Geyer, C. J. (1994). Discussion of the paper by Tierny. *Annals of Statistics* **22** 1747–1758.

Chen, H.-F. (2002). *Stochastic Approximation and Its Applications.* Kluwer Academic Publishers, Dordrecht.

Fletcher, R. (1987). *Practical Methods of Optimization*, Second ed. John Wiley & Sons.

Geyer, C. J. (1990). Likelihood and Exponential Families PhD thesis, University of Washington.

Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society,* Series B **56** 261-274.

Geyer, C. J. (2009a). Likelihood Inference in Exponential Families and Directions of Recession. *Electronic Journal of Statistics* **3** 259–289.

Geyer, C. J. (2009b). `mcmc`: Markov chain Monte Carlo. R pakage version 0.7-3.

Geyer, C. J. (2010). `aster2`: Aster models. R pakage version 0.1.

Geyer, C. J. (2011). Introduction to MCMC. In *Handbook of Markov Chain Monte Carlo* (S. P. Brooks, A. E. Gelman, G. L. Jones and X. L. Meng, eds.) Chapman & Hall/CRC, Boca Raton, FL.

Geyer, C. J. and Johnson, L. T. (2010). `potts`: Markov chain Monte Carlo for Potts Models. R package version 0.4.

Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo Maximum Likelihood for Dependent Data. *Journal of the Royal Statistical Society,* Series B **54** 657-699.

Goodreau, S. M. (2007). Advances in Exponential Random Graph (p*) Models Applied to a Large Social Network. *Social Networks* **29** 231–248.

Goodreau, S. M., Kitts, J. A. and Morris, M. (2009). Birds of a Feather, or Friend of a Friend? Using Exponential Random Graph Models to Investigate Adolescent Social Networks. *Demography* **46** 103—125.

Gu, M. G. and Zhu, H.-T. (2001). Maximum Likelihood Estimation for Spatial Models by Markov Chain Monte Carlo Stochastic Approximation. *Journal of the Royal Statistical Society,* Series B **63** 339–355.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M. and Morris, M. (2003). `statnet`: Software Tools for the Statistical Modeling of Network Data. Version 2.0. Project home page at `http://statnetproject.org`.

Hummel, R., Hunter, D. R. and Handcock, M. S. (2010). A Steplength Algorithm for Fitting ERGMs Technical Report No. 10-03, Pennsylvania State University.

Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M. and Morris, M. (2008). ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks. *Journal of Statistical Software* **24**.

Ising, E. (1925). Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei* **31** 253–258.

Jaynes, E. T. (1978). Where Do We Stand on Maximum Entroy? In *The Maximum Entropy Formalism* (R. D. Levine and M. Tribus, eds.) Cambridge: Massassachusetts Institute of Technology Press.

Jones, G. L. (2004). On the Markov chain Central Limit Theorem. *Probability Surveys* **1** 299–320.

Kushner, H. J. and Yin, G. G. (1997). *Stochastic Approximation Algorithms and Applications*. Springer, New York.

Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation*, Second ed. Springer.

Lehmann, E. L. and Romano, J. P. (2005). *Testing Statistical Hypotheses*, 3rd ed. Springer.

Liang, F. (2010). Trajectory Averaging for Stochastic Approximation MCMC Algorithms. *The Annals of Applied Statistics* **38** 2823–2856.

Moyeed, R. A. and Baddeley, A. J. (1991). Stochastic Approximation of the MLE for a Spatial Point Pattern. *Scandinavian Journal of Statistics* **18** 39–50.

Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*, First ed. Springer.

Okabayashi, S. (2011). Parameter Estimation in Social Network Models PhD thesis, University of Minnesota.

Okabayashi, S., Johnson, L. and Geyer, C. J. (2011). Extending Pseudo-likelihood for Potts Models. *Statistica Sinica* **21** 331–347.

Penttinen, A. (1984). Modelling interactions in spatial point patterns: parameter estimation by the maximum likelihood method. *Jyväskylä Studies in Computer Science, Economics and Statistics* **7**.

Potts, R. B. (1952). Some Generalized Order-Disorder Transformations. *Proceedings of the Cambridge Philosphical Society* **48** 106–109.

Rinaldo, A., Fienberg, S. E. and Zhou, Y. (2009). On the geometry of discrete Exponential Families with Application to Exponential Random Graph Models. *Electronic Journal of Statistics* **3** 446–484.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *Annals of Mathematical Statistics* **22** 400–407.

Roberts, G. O. and Rosenthal, J. S. (1997). Geometric ergodicity and hybrid Markov chains. *Electronic Communications in Probability* **2** 13–25.

Roberts, G. O. and Rosenthal, J. S. (2004). General State space Markov chains and MCMC algorithms. *Probability Surveys* **1** 20–71.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.

Rockafellar, R. T. and Wets, R. J.-B. (2004). *Variational Analysis. corrected second printing*. Springer-Verlag, Berlin.

Saul, Z. M. and Filkov, V. (2007). Exploring biological network structure using exponential random graph models. *Bioinformatics* **23** 2604-02611.

Shaw, R. G., Geyer, C. J., Wagenius, S., Hangelbroek, H. H. and Etterson, J. R. (2008). Unifying Life-History Analyses for Inference of Fitness and Population Growth. *The American Naturalist* **172** E35-E47.

Snijders, T. A. B. (2002). Markov Chain Monte Carlo Estimation of Exponential Random Graph Models. *Journal of Social Structure* **3**.

Strauss, D. and Ikeda, M. (1990). Pseudolikelihood Estimation for Social Networks. *Journal of the American Statistical Association* **85** 204-212.

Sun, W. and Yuan, Y.-X. (2006). *Optimization Theory and Methods: Nonlinear Programming.* Springer.

Swendsen, R. H. and Wang, J.-S. (1987). Nonuniversal Critical Dynamics in Monte Carlo simulations. *Physics Review Letters* **58** 86-88.

van Duijn, M. A. J., Gile, K. J. and Handcock, M. S. (2009). A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks* **31** 52-62.

Wang, J. S. and Swendsen, R. H. (1990). Cluster Monte Carlo algorithms. *Physics A* **167** 565–579.

Wasserman, S. and Pattison, P. (1996). Logit Models and Logistic Regression for Social Networks: I. An Introduction to Markov Graphs and p*. *Psychometrika* **61** 401-425.

Younes, L. (1988). Estimation and Annealing for Gibbsian Fields. *Ann. Inst. Henri Poincare* **24** 269–294.

Younes, L. (1989). Parametric Inference for Imperfectly Observed Gibbsian Fields. *Probability Theory and Related Fields* **82** 625–645.