Mach Learn (2007) 69: 1–33
DOI 10.1007/s10994-007-5022-x

Surrogate maximization/minimization algorithms and extensions

Zhihua Zhang · James T. Kwok · Dit-Yan Yeung

Received: 28 November 2005 / Revised: 11 July 2007 / Accepted: 9 August 2007 /
Published online: 19 September 2007
Springer Science+Business Media, LLC 2007

Abstract Surrogate maximization (or minimization) (SM) algorithms are a family of algorithms that can be regarded as a generalization of expectation-maximization (EM) algorithms. An SM algorithm aims at turning an otherwise intractable maximization problem into a tractable one by iterating two steps. The S-step computes a tractable surrogate function to substitute the original objective function and the M-step seeks to maximize this surrogate function. Convexity plays a central role in the S-step. SM algorithms enjoy the same convergence properties as EM algorithms. There are mainly three approaches to the construction of surrogate functions, namely, by using Jensen's inequality, first-order Taylor approximation, and the low quadratic bound principle. In this paper, we demonstrate the usefulness of SM algorithms by taking logistic regression models, AdaBoost and the log-linear model as examples. More specifically, by using different surrogate function construction methods, we devise several SM algorithms, including the standard SM, generalized SM, gradient SM, and quadratic SM algorithms, and their two variants called the conditional surrogate maximization (CSM) and surrogate conditional maximization (SCM) algorithms.

Keywords Surrogate function · Convexity · Logistic regression · AdaBoost · Log-linear model

1 Introduction

In machine learning and statistics, optimization plays a very important role because many problems require performing maximization or minimization of some objective function.

Editor: Dale Schuurmans.

Z. Zhang
Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720, USA

J.T. Kwok · D.-Y. Yeung (✉)
Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
e-mail: dyyeung@cse.ust.hk

One widely used objective function is the log-likelihood function. Since it is closely related to convex (or concave) functions (Rockafellar 1970), convexity (or concavity) also plays a central role in such problems. A successful example is the well-known expectation-maximization (EM) algorithm (Dempster et al. 1977). Becker et al. (1997) and Lange et al. (2000) showed that the EM algorithm can be derived from either Jensen's inequality or the concavity property of the log function. Along this line, a family of EM-like algorithms without missing data (Becker et al. 1997) have been devised to handle cases involving no missing data. Lange et al. (2000) unified this family of algorithms under the framework of the so-called *optimization transfer* algorithms, in which all algorithms rely on optimizing a function that serves as a surrogate for the original objective function. By invoking convexity arguments, a general principle providing guidelines on constructing these surrogate functions, as well as some specific examples, have been discussed (Lange et al. 2000). Depending on the context, this often relies on three important tools, namely, *Jensen's inequality*, *first-order Taylor approximation*, and the *low quadratic bound principle*.

Optimization transfer algorithms are very efficient because they can make an otherwise hard or very complicated optimization problem simpler. For example, an optimization transfer algorithm can decouple the correlation among parameters so that they can be estimated in parallel. It can also locally linearize a convex function near some value so as to make the problem at hand tractable. It can avoid the computational problem of inverting large matrices as required by Newton's method. Moreover, optimization transfer enjoys the same local convergence properties as standard EM.

Other names have been used for optimization transfer methods. In the context of multidimensional scaling (MDS) (Borg and Groenen 1997), optimization transfer is referred to as *iterative majorization*; while in convex optimization (Boyd and Vandenberghe 2004), it is usually called the *auxiliary function* method. To contrast optimization transfer methods with the standard EM algorithm for missing data problems, Meng (2000) suggested using SM algorithm as the alternative name for optimization transfer. Here, "*S*" stands for the surrogate step while "*M*" stands for the maximization (or minimization, depending on the optimization problem at hand) step. In this paper we also prefer the name "SM algorithms" as it reflects more accurately the spirit of this family of algorithms. Like EM algorithms, SM algorithms are also gaining popularity in computational statistics. However, although EM algorithms are commonly used in machine learning nowadays, this is not the case for SM algorithms. This paper attempts to demonstrate the power and potential of SM algorithms in machine learning, by using generalized linear models, such as logistic regression and log-linear models, as specific examples for illustration.

We address two major issues in devising SM algorithms, namely, how a surrogate function is defined and how the resultant surrogate function is maximized. On the first problem, there exist three main approaches, namely, by using Jensen's inequality, first-order Taylor approximation, and the low quadratic bound principle. The first two approaches follow readily from the properties of convex functions, while the third one uses a quadratic function to approximate the original objective function. On the second problem, in general different maximization methods are required for different surrogate functions. This leads to the standard SM, generalized SM, gradient SM, and quadratic SM algorithms, and their two variants called the conditional surrogate maximization (CSM) and surrogate conditional maximization (SCM) algorithms (Meng 2000).

1.1 Contributions

To demonstrate how the three approaches mentioned above can be used to construct a surrogate function, we consider the optimization problem corresponding to the binary logistic regression model. Based on Jensen's inequality, we decouple the correlation among the estimated parameters and decompose the original high-dimensional optimization problem into a set of one-dimensional sub-problems which can then be handled separately. Although we cannot obtain a one-step closed-form iterative procedure, we present a gradient SM algorithm by borrowing ideas from the gradient EM algorithm (Lange 1995). Moreover, we show that the iterative procedure of (Collins et al. 2002) can be regarded as a generalized SM algorithm analogous to the generalized EM algorithm (Dempster et al. 1977). Based on the first-order Taylor approximation, we express the original objective function as the difference of two convex functions (i.e., a convex function plus a concave function), leading to a quadratic surrogate function. Based on the low quadratic bound principle (Böhning and Lindsay 1988), we devise quadratic SM algorithms. The essence of quadratic SM algorithms is to approximate the Hessian matrix in the pure Newton method with a simpler positive semidefinite matrix, and we will adopt a constant matrix in our case here. Thus, we only have to compute the inverse of the Hessian matrix just once and inversion of large matrices at each iteration can be avoided.

While Lange et al. (2000) also used these three approaches to construct their SM algorithms, they considered different approaches for different optimization problems. In contrast, we consider the use of the three approaches for the same optimization problem. Thus, our treatment allows us to show that different construction approaches can be used to devise different SM algorithms for the same optimization problem. In addition, based on combinations of Jensen's inequality, first-order Taylor approximation and the low quadratic bound principle, we present the fourth approach for constructing surrogate functions. Quite surprisingly, the SM algorithms obtained turn out to be equivalent to the parallel Bregman distance algorithms of (Collins et al. 2002), and thus our method can be seen as providing a new derivation for their algorithms. Compared with (Della Pietra et al. 2001) and (Collins et al. 2002), the mathematical skills required for our approach are much simpler because we only need to utilize Jensen's inequality or first-order Taylor approximation over a convex function.

Our other contributions are to devise CSM and SCM algorithms for the binary logistic regression model, and SM algorithms for multi-class logistic regression models and AdaBoost. More importantly, our approaches naturally guarantee convergence of the corresponding iterative algorithms. Moreover, we derive an SM algorithm for the log-linear model. On the one hand, this illustrates an application of the SM algorithm to a constrained optimization problem. On the other hand, our SM algorithm may be seen as an amendment of the generalized iterative scaling (GIS) (Darroch and Ratcliff 1972) as the constraint in GIS is not exactly satisfied. In summary, we believe that SM algorithms can find wide applications in machine learning even beyond generalized linear models.

1.2 Related work

The idea behind SM algorithms has been used in logistic regression models and AdaBoost (see, e.g., Minka 2003; Lebanon and Lafferty 2001). However, our work has been mainly motivated by some recent works (Collins et al. 2002; Kivinen and Warmuth 1999; Lafferty 1999) which are based on Bregman distance optimization methods. Simply put, the Bregman distance between two vectors is defined via a convex function on a convex set that contains these two vectors. Della Pietra et al. (1997) applied Bregman distance optimization to log-linear models, while Della Pietra et al. (1997) and Collins et al. (2002) discussed its relationship with GIS for log-linear models (Darroch and Ratcliff 1972). Like GIS, the core spirit of Bregman distance optimization is from convex analysis (Rockafellar 1970).

However, this approach requires considerable mathematical skills to construct a Bregman function that matches the problem in question. Furthermore, in order to use Bregman distance optimization, it is common to reformulate the unconstrained optimization problem as an equivalent constrained optimization problem subject to some constraints. This makes the problem much more technically involved. Della Pietra et al. (2001) also recognized these difficulties and sought to use the Legendre transformation technique (Rockafellar 1970). The main difference between (Della Pietra et al. 2001) and (Collins et al. 2002) is that the former works with the argument at which a convex conjugate takes on its value, while the latter works with the value of the functional itself. This makes it more natural to formulate a duality theorem.

The rest of this paper is organized as follows. Sect. 2 presents the generic principle of SM algorithms and Sect. 3 presents two extensions of SM algorithms. This is then applied to the binary logistic regression model in Sects. 4 and 5. In Sects. 6–8, we further present SM algorithms for the multi-class logistic regression model, AdaBoost, and the log-linear model, respectively. The last section gives some concluding remarks.

2 Generic structure of SM algorithms

In many applications we have to consider the problem of maximizing an arbitrary function $L(\boldsymbol{\theta})$ with respect to (w.r.t.) some parameter vector $\boldsymbol{\theta} \in \mathbb{R}^q$. Given an estimate $\boldsymbol{\theta}(t)$ at the tth iteration, a typical SM algorithm (Lange et al. 2000; Meng 2000) consists of the following two steps:

Surrogate Step (S-Step): Substitute $L(\boldsymbol{\theta})$ by a surrogate function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$, such that

$$L(\boldsymbol{\theta}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t)) \tag{1}$$

for all $\boldsymbol{\theta}$, with equality holding at $\boldsymbol{\theta} = \boldsymbol{\theta}(t)$.

Maximization Step (M-Step): Obtain the next parameter estimate $\boldsymbol{\theta}(t+1)$ by maximizing the surrogate function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$ w.r.t. $\boldsymbol{\theta}$, i.e.,

$$\boldsymbol{\theta}(t+1) = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t)). \tag{2}$$

Note that the SM algorithms can be applied equally well to the minimization of $L(\boldsymbol{\theta})$, by simply reversing the inequality sign in (1) and changing the “max” to “min” in (2). Therefore, in the sequel, “M” stands for either maximization or minimization depending on the optimization problem at hand.

Depending on the surrogate functions obtained, different SM algorithms can be devised accordingly. In the *standard SM algorithm*, a closed-form solution for $\boldsymbol{\theta}(t+1)$ exists in the M-step. However, it is not always possible to obtain a closed-form solution for $\boldsymbol{\theta}(t+1)$ in the M-step. In the same spirit as the generalized EM algorithm (Dempster et al. 1977), we can devise a *generalized SM algorithm*, where, instead of maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$, we only attempt to find a $\boldsymbol{\theta}(t+1)$ such that $Q(\boldsymbol{\theta}(t+1)|\boldsymbol{\theta}(t)) \geq Q(\boldsymbol{\theta}(t)|\boldsymbol{\theta}(t))$. Alternatively, in the same spirit as the gradient EM algorithm (Lange 1995), we may also devise a *gradient SM algorithm*, as

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - (\nabla^2 Q(\boldsymbol{\theta}(t)|\boldsymbol{\theta}(t)))^{-1} \nabla Q(\boldsymbol{\theta}(t)|\boldsymbol{\theta}(t)),$$

which is indeed the pure Newton method over $Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$ instead of $L(\boldsymbol{\theta})$ because $L(\boldsymbol{\theta}) - Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$ has a stationary point at $\boldsymbol{\theta} = \boldsymbol{\theta}(t)$ so that $\nabla L(\boldsymbol{\theta}(t)) = \nabla Q(\boldsymbol{\theta}(t)|\boldsymbol{\theta}(t))$.

2.1 Convergence properties

Let $\Omega \subseteq \mathbb{R}^q$ be a set of feasible parameter values and

$$L : \boldsymbol{\theta} \in \Omega \mapsto L(\boldsymbol{\theta}) \in \mathbb{R}$$

defines the objective function to be maximized. We regard each SM iteration as a point-to-set mapping A such that $\boldsymbol{\theta}(t)$ becomes $\boldsymbol{\theta}(t+1) \in A(\boldsymbol{\theta}(t))$. That is, the generalized SM algorithm leads us to the following problem

$$\text{Find } \hat{\boldsymbol{\theta}} \in \Omega \text{ such that } L(\hat{\boldsymbol{\theta}}) \geq L(\boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta} \in \Omega.$$

Given an initial value $\boldsymbol{\theta}(0)$, we can generate an iterative sequence $\{\boldsymbol{\theta}(t)\}$ such that $\boldsymbol{\theta}(t+1) \in A(\boldsymbol{\theta}(t))$. It follows from the definition of the standard (or generalized) SM algorithm that

$$L(\boldsymbol{\theta}(t+1)) \geq Q(\boldsymbol{\theta}(t+1)|\boldsymbol{\theta}(t)) \geq Q(\boldsymbol{\theta}(t)|\boldsymbol{\theta}(t)) = L(\boldsymbol{\theta}(t)).$$

Let $\{L(\boldsymbol{\theta}(t))\}$ be bounded above. Then $L(\boldsymbol{\theta}(t))$ converges monotonically to some $L^* < \infty$.

The standard (generalized) SM algorithm enjoys the same convergence properties (Dempster et al. 1977; Wu 1983) as the standard (generalized) EM algorithm. Throughout this subsection, we make the following assumptions:

$$L \text{ is continuous in } \Omega \text{ and differentiable in the interior of } \Omega, \tag{3}$$

$$\Omega_0 = \{\boldsymbol{\theta} \in \Omega : L(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}(0))\} \text{ is compact for any } L(\boldsymbol{\theta}(0)) > -\infty, \tag{4}$$

$$Q(\boldsymbol{\theta}|\boldsymbol{\phi}) \text{ is continuous in both } \boldsymbol{\theta} \text{ and } \boldsymbol{\phi} \text{ in } \Omega, \text{ and differentiable in } \boldsymbol{\theta} \text{ in the interior of } \Omega. \tag{5}$$

From the convergence results in (Wu 1983, Theorems 2 and 3), it is straightforward to obtain the convergence results to our generalized SM (standard) algorithm. Specifically, let $\mathcal{M}$ and $\mathcal{S}$ be the set of local maxima and the set of stationary points, respectively, of L in the interior of Ω. The condition that $Q(\boldsymbol{\theta}|\boldsymbol{\phi})$ is continuous in both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ in Ω is a sufficient condition for that A is a closed point-to-set mapping over the complement of $\mathcal{S}$ ($\mathcal{M}$). Since $L(\boldsymbol{\theta}) - Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$ has a stationary point at $\boldsymbol{\theta} = \boldsymbol{\theta}(t)$, we have $\nabla L(\boldsymbol{\theta}(t)) = \nabla Q(\boldsymbol{\theta}(t)|\boldsymbol{\theta}(t)) \neq 0$ for any $\boldsymbol{\theta}(t) \notin \mathcal{S}$. This implies that $\boldsymbol{\theta}(t)$ is not a local maximum of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$ over $\boldsymbol{\theta} \in \Omega$. From the definition of the M-step, we have $Q(\boldsymbol{\theta}(t+1)|\boldsymbol{\theta}(t)) > Q(\boldsymbol{\theta}(t)|\boldsymbol{\theta}(t))$, hence $L(\boldsymbol{\theta}(t+1)) > L(\boldsymbol{\theta}(t))$ for all $\boldsymbol{\theta}(t) \notin \mathcal{S}$. Therefore, it follows from Zangwill's global convergence theorem (Wu 1983) that

Theorem 1 *Suppose that the conditions* (3), (4) *and* (5) *are satisfied. Then all the limit points of any iterative sequence* $\{\boldsymbol{\theta}(t)\}$ *of a generalized SM algorithm are stationary points of* $L(\boldsymbol{\theta})$ *and* $L(\boldsymbol{\theta}(t))$ *converges monotonically to* $L(\boldsymbol{\theta}^*)$ *for some stationary point* $\boldsymbol{\theta}^*$. *Furthermore, if* Q *also satisfies*

$$\sup_{\boldsymbol{\theta} \in \Omega} Q(\boldsymbol{\theta}|\hat{\boldsymbol{\phi}}) > Q(\hat{\boldsymbol{\phi}}|\hat{\boldsymbol{\phi}}) \quad \textit{for any } \hat{\boldsymbol{\phi}} \in \mathcal{S} \backslash \mathcal{M}, \tag{6}$$

then all the limit points of any sequence $\{\boldsymbol{\theta}(t)\}$ *of the SM algorithm are local maxima of* $L(\boldsymbol{\theta})$ *and* $L(\boldsymbol{\theta}(t))$ *converges monotonically to* $L(\boldsymbol{\theta}^*)$ *for some local maximum* $\boldsymbol{\theta}^*$.

Condition (5) is in fact very weak as it is usually satisfied in most practical cases. For example, this condition always holds in Sects. 4–8. Condition (6) is typically hard to verify. However, if $L(\boldsymbol{\theta})$ is concave in $\boldsymbol{\theta}$ and bounded above ($< \infty$), then $L(\boldsymbol{\theta})$ has a unique stationary point which is the global maximum. Thus, we have the following theorem.

Theorem 2 *Suppose that the conditions* (3), (4) *and* (5) *are satisfied. If* $L(\boldsymbol{\theta})$ *is concave in* $\boldsymbol{\theta}$ *and bounded above, then the limit point of any sequence* $\{\boldsymbol{\theta}(t)\}$ *of a generalized SM algorithm is the global maximum of* $L(\boldsymbol{\theta})$ *and* $L(\boldsymbol{\theta}(t))$ *converges monotonically to* $L(\boldsymbol{\theta}^*)$ *for the global maximum* $\boldsymbol{\theta}^*$.

2.2 Construction of surrogate functions

Clearly, construction of the surrogate function is key to SM algorithms in turning an otherwise intractable optimization problem into a tractable one. On the one hand, the closer is the surrogate function to $L(\boldsymbol{\theta})$, the more efficient is the SM algorithm. On the other hand, a good surrogate function should preferably have a closed-form solution in the M-step. Lange et al. (2000) discussed some general principles and presented three methods for the design of surrogate functions in which convexity of functions plays a central role.

Suppose a function $f : \mathcal{S} \to (-\infty, +\infty]$ is convex on a closed convex set $\mathcal{S} \subseteq \mathbb{R}^q$. The first method stems from Jensen's inequality

$$f\left(\sum_{i=1}^{k} \alpha_i \mathbf{u}_i\right) \leq \sum_{i=1}^{k} \alpha_i f(\mathbf{u}_i),$$

where $\alpha_i \geq 0$ $(i = 1, \ldots, k)$ and $\sum_{i=1}^{k} \alpha_i = 1$, or its variant

$$f\left(\sum_{i=1}^{k} \alpha_i \mathbf{u}_i\right) \leq \sum_{i=1}^{k} \alpha_i f(\mathbf{u}_i) + \left(1 - \sum_{i=1}^{k} \alpha_i\right) f(\mathbf{0}),$$

where $\sum_{i=1}^{k} \alpha_i \leq 1$.

The following two extensions of Jensen's inequality are also useful. The first one is

$$f(\mathbf{c}^T \mathbf{u}) \leq \sum_i \frac{c_i w_i}{\mathbf{c}^T \mathbf{w}} f\left(\frac{\mathbf{c}^T \mathbf{w}}{w_i} u_i\right),$$

where all elements of $\mathbf{c} = [c_i]$ and $\mathbf{w} = [w_i]$ are positive, while the second one is

$$f\left(\sum_{i=1}^{k} c_i \mathbf{u}_i\right) \leq \sum_{i=1}^{k} \alpha_i f\left(\frac{c_i}{\alpha_i}(\mathbf{u}_i - \mathbf{v}_i) + \sum_{j=1}^{k} c_j \mathbf{v}_j\right),$$

where $\alpha_i \geq 0$ $(i = 1, \ldots, k)$ and $\sum_{i=1}^{k} \alpha_i = 1$, and $\alpha_i > 0$ whenever $c_i \neq 0$ (De Pierro 1995). These inequalities can be used to decouple the correlation among the $\mathbf{u}_i$'s.

The second construction method makes use of the following property: When $f(\cdot)$ is also differentiable on its domain $\mathcal{S}$, it can be linearized by first-order Taylor approximation, as

$$f(\mathbf{u}) \geq f(\mathbf{v}) + \nabla f(\mathbf{v})^T (\mathbf{u} - \mathbf{v}), \quad \text{for } \mathbf{u}, \mathbf{v} \in \mathcal{S}.$$

Since most continuous functions can be expressed as the difference of two convex functions, we can often use this trick to construct a surrogate function. For example, if for any $f(\mathbf{u}) =$

$g(\mathbf{u}) - h(\mathbf{u})$ where both $g(\mathbf{u})$ and $h(\mathbf{u})$ are convex, we can write $f(\mathbf{u}) \leq g(\mathbf{u}) - h(\mathbf{v}) - \nabla h(\mathbf{v})^T(\mathbf{u}-\mathbf{v})$. The use of differences of convex (d.c.) functions is a very important strategy in convex optimization and has received much attention recently in machine learning. For example, the recently proposed convex-concave computational procedure (CCCP) (Yuille and Rangarajan 2001) is essentially based on this strategy.

The third method uses the low quadratic bound principle (Böhning and Lindsay 1988). Suppose there exists a $\mathbf{u}$-independent positive semidefinite matrix $\mathbf{B}$ such that $\mathbf{B} - \nabla^2 f(\mathbf{u})$ is positive semidefinite. Then, it can be shown that

$$f(\mathbf{u}) \leq f(\mathbf{v}) + \nabla f(\mathbf{v})^T(\mathbf{u}-\mathbf{v}) + \frac{1}{2}(\mathbf{u}-\mathbf{v})^T\mathbf{B}(\mathbf{u}-\mathbf{v}).$$

This is often used to define a quadratic surrogate function that can avoid the inversion of the Hessian matrix in Newton's method.

3 Extensions of SM: CSM and SCM

For a multi-parameter optimization problem with a set of parameter vectors $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C\}$, the objective function $L(\Theta)$ may also be expressed as $L(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C)$. In order to maximize $L(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C)$ w.r.t. $\boldsymbol{\theta}_i$'s, we use the so-called *block relaxation* diagram proposed by (De Leeuw 1994). For simplicity of notation, let $L_i = L(\boldsymbol{\theta}_1(*), \ldots, \boldsymbol{\theta}_{i-1}(*), \boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}(t), \ldots, \boldsymbol{\theta}_C(t))$, where all $*$'s are simultaneously either t or $t+1$. The block relaxation algorithm obtains $\boldsymbol{\theta}_i(t+1)$ by maximizing L_i. If $* = t$, the procedure is called *parallel-update* (corresponding to the *Jacobi method* in numerical mathematics), otherwise it is called *sequential-update* (corresponding to the *Gauss–Seidel method*).

Instead of working with $L(\Theta)$ directly, we apply the SM algorithm to the maximization of L_i w.r.t. $\boldsymbol{\theta}_i$, i.e., we first for L_i's define surrogate functions $Q_i(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i(t))$, whose types can be different for different L_i's, and then maximize $Q_i(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i(t))$. In many cases, since L_i is in fact a log-likelihood function conditioned on $\boldsymbol{\theta}_l$'s ($l \neq i$) in computational statistics, we refer to $Q_i(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i(t))$ as a conditional surrogate function. As a result, this variant of the SM algorithm is called the *conditional surrogate maximization* (CSM) algorithm (Table 1). It is worth noting that the CSM algorithm is closely related to the CEM algorithm (Jebara and Pentland 1998), which is for maximizing an approximate conditional likelihood function in mixture models.

An alternative to dealing with multiple variables (parameters) is based on the idea behind the ECM algorithm (Meng and Rubin 1993), where one first computes the E-step and then

Table 1 Block relaxation diagram of the CSM algorithm

Begin	Start with $\boldsymbol{\theta}_i(0) \in \mathbb{R}^m$ for $i = 1, \ldots, C$ and $t = 0$
S-step $(t+1).1$	Define a surrogate function $Q_1(\boldsymbol{\theta}_1|\boldsymbol{\theta}_1(t))$ for L_1.
M-step $(t+1).1$	Find a $\boldsymbol{\theta}_1(t+1)$ such that $Q_1(\boldsymbol{\theta}_1(t+1)|\boldsymbol{\theta}_1(t)) \geq Q_1(\boldsymbol{\theta}_1(t)|\boldsymbol{\theta}_1(t))$.
S-step $(t+1).2$	Define a surrogate function $Q_2(\boldsymbol{\theta}_2|\boldsymbol{\theta}_2(t))$ for L_2.
M-step $(t+1).2$	Find a $\boldsymbol{\theta}_2(t+1)$ such that $Q_2(\boldsymbol{\theta}_2(t+1)|\boldsymbol{\theta}_2(t)) \geq Q_2(\boldsymbol{\theta}_2(t)|\boldsymbol{\theta}_2(t))$.
...	...
S-step $(t+1).C$	Define a surrogate function $Q_C(\boldsymbol{\theta}_C|\boldsymbol{\theta}_C(t))$ for L_C.
M-step $(t+1).C$	Find a $\boldsymbol{\theta}_C(t+1)$ such that $Q_C(\boldsymbol{\theta}_C(t+1)|\boldsymbol{\theta}_C(t)) \geq Q_C(\boldsymbol{\theta}_C(t)|\boldsymbol{\theta}_C(t))$.
Motor	If not converged, then $t \leftarrow t+1$ and go to S-step $(t+1).1$.

Table 2 Block relaxation diagram of the SCM algorithm

Begin	Start with $\boldsymbol{\theta}_i(0) \in \mathbb{R}^m$ for $i = 1, \ldots, C$ and $t = 0$.
S-step t	Define a surrogate function $Q(\Theta\|\Theta(t))$ for $L(\Theta)$.
M-step $t.1$	Find a $\boldsymbol{\theta}_1(t+1)$ that satisfies $Q(\Theta\|\Theta(t)) \geq Q(\Theta(t)\|\Theta(t))$ subject to $r_1(\boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_C) = r_1(\boldsymbol{\theta}_2(t), \ldots, \boldsymbol{\theta}_C(t))$.
M-step $t.2$	Find a $\boldsymbol{\theta}_2(t+1)$ that satisfies $Q(\Theta\|\Theta(t)) \geq Q(\Theta(t)\|\Theta(t))$ subject to $r_2(\boldsymbol{\theta}_1, \boldsymbol{\theta}_3, \ldots, \boldsymbol{\theta}_C) = r_2(\boldsymbol{\theta}_1(*), \boldsymbol{\theta}_3(t), \ldots, \boldsymbol{\theta}_C(t))$.
...	...
M-step $t.C$	Find a $\boldsymbol{\theta}_C(t+1)$ that satisfies $Q(\Theta\|\Theta(t)) \geq Q(\Theta(t)\|\Theta(t))$ subject to $r_C(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{C-1}) = r_C(\boldsymbol{\theta}_1(*), \boldsymbol{\theta}_2(*), \ldots, \boldsymbol{\theta}_{C-1}(*))$.
Motor	If not converged, then $t \leftarrow t+1$ and go to S-step t.

decomposes the M-step into several CM-steps. Analogous to the setting of ECM, we also propose a *surrogate conditional maximization* (SCM) algorithm (Table 2). The differences between CSM and SCM can be clearly seen from Tables 1 and 2. Specifically, CSM decomposes each SM-step into C conditional SM-steps, while SCM only decomposes each M-step of SM into C conditional M-steps. In Table 2, $r_i(\Theta)$ is a vector function of Θ. Specifically, $r_i(\Theta) = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_C)$ which is a vector containing all the parameters except $\boldsymbol{\theta}_i$.

4 SM algorithms for binary logistic regression model

In this section we focus on parameter estimation in the binary logistic regression model and present several SM algorithms based on using different methods for constructing the surrogate function. The first is based on Jensen's inequality (Sect. 4.1), the second is based on the first-order Taylor approximation (Sect. 4.2), the third is based on the low quadratic bound principle (Sect. 4.3), while the last one is based on a combination of approaches (Sect. 4.4). Moreover, we will also see that the generalized SM algorithm is equivalent to the parallel Bregman optimization algorithm in (Collins et al. 2002).

Let $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be a finite set of training examples, where each instance $\mathbf{x}_i$ from a domain or instance space $\mathcal{X}$ corresponds to a label $y_i \in \{-1, +1\}$. We also assume that we are given a set of real-valued feature functions, $h_1, \ldots, h_m$, on $\mathcal{X}$. Now our goal is to label the $\mathbf{x}_i$'s using a linear combination of these features. In other words, we want to find a parameter vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)^T \in \mathbb{R}^m$ such that $f_\lambda(\mathbf{x}_i) = \sum_{j=1}^m \lambda_j h_j(\mathbf{x}_i)$ is a good approximation of the underlying classification function. Instead of using f_λ directly as a classification rule, we usually postulate that the y_i's come from a probabilistic model associated with $f_\lambda(\mathbf{x}_i)$. In logistic regression models, one suggestion is that the posterior probability of y_i is given by a logistic function of $f_\lambda(\mathbf{x}_i)$, as

$$\hat{p}(y_i|\mathbf{x}_i, \boldsymbol{\lambda}) = \frac{1}{1 + \exp\{-y_i \sum_{j=1}^m \lambda_j h_j(\mathbf{x}_i)\}}. \tag{7}$$

Accordingly, we can use the maximum likelihood estimation method for $\boldsymbol{\lambda}$. Here we reformulate maximum likelihood estimation as an equivalent minimization problem, which is based on the following loss function

$$L_b(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \ln\left\{1 + \exp\left(-y_i \sum_{j=1}^{m} \lambda_j h_j(\mathbf{x}_i)\right)\right\}.$$

This problem was also addressed by an algorithm called LogitBoost (Collins et al. 2002) in the context of boosting (Friedman et al. 2000; Schapire 1990).

Let us define

$$g_{ij} = -y_i h_j(\mathbf{x}_i) \tag{8}$$

and $\mathbf{g}_i = (g_{i1}, \ldots, g_{im})^T$. Thus,

$$L_b(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \ln\left\{1 + \exp\left(\sum_{j=1}^{m} \lambda_j g_{ij}\right)\right\}. \tag{9}$$

As in (Collins et al. 2002), we assume that

$$\sum_{j=1}^{m} |g_{ij}| \leq 1. \tag{10}$$

Moreover, without loss of generality, we assume throughout this paper that $g_{ij} \neq 0$ for all i and j. If there exists some $g_{ij} = 0$, we can simply remove the corresponding term from the summation in $\exp\{\sum_{j=1}^{m} \lambda_j g_{ij}\}$ so that the same results are still applicable.

4.1 Using Jensen's inequality

We rewrite $L_b(\boldsymbol{\lambda})$ in (9) as

$$L_b(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \ln\left\{1 + \exp\left[\sum_{j=1}^{m} |g_{ij}| \left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)) + \boldsymbol{\lambda}(t)^T \mathbf{g}_i\right) + (1 - \alpha_i)\boldsymbol{\lambda}(t)^T \mathbf{g}_i\right]\right\},$$

where

$$\alpha_i = \sum_{j=1}^{m} |g_{ij}|. \tag{11}$$

Since $\frac{d^2 \ln(1+\exp(u))}{du^2} = \frac{\exp(u)}{(1+\exp(u))^2} > 0$, $\ln(1 + \exp(\cdot))$ is convex, and hence

$$\begin{aligned} L_b(\boldsymbol{\lambda}) &\leq \sum_{i=1}^{n} (1 - \alpha_i) \ln(1 + \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)) \\ &\quad + \sum_{i=1}^{n} \left\{\sum_{j=1}^{m} |g_{ij}| \ln\left[1 + \exp\left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)) + \boldsymbol{\lambda}(t)^T \mathbf{g}_i\right)\right]\right\} \\ &\equiv Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)). \end{aligned} \tag{12}$$

It is easy to show that $Q_z(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)) = L_b(\boldsymbol{\lambda}(t))$. Hence, $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ can be used as a surrogate function of $L_b(\boldsymbol{\lambda})$. We then minimize $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ w.r.t. the λ_j's, by setting the partial derivative

$$\frac{\partial Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))}{\partial \lambda_j} = \sum_{i=1}^{n} g_{ij} \frac{\exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i + \frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)))}{1 + \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i + \frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)))}$$

to zero. However, a closed-form solution cannot be found. There are two methods to tackle this problem. One is to employ a strategy similar to the generalized EM algorithm (Dempster et al. 1977), leading to a generalized SM algorithm. Alternatively, we can resort to a gradient SM algorithm analogous to the gradient EM algorithm (Lange 1995). Here, we employ this strategy. Using

$$\left.\frac{\partial Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))}{\partial \lambda_j}\right|_{\lambda_j=\lambda_j(t)} = \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t)) g_{ij},$$

$$\left.\frac{\partial^2 Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))}{\partial \lambda_j^2}\right|_{\lambda_j=\lambda_j(t)} = \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t))(1 - p_i(\boldsymbol{\lambda}(t)))|g_{ij}|,$$

where $p_i(\boldsymbol{\lambda}) = \frac{\exp(\boldsymbol{\lambda}^T \mathbf{g}_i)}{1+\exp(\boldsymbol{\lambda}^T \mathbf{g}_i)}$, we update the current parameter estimate $\lambda_j(t)$ to

$$\lambda_j(t+1) = \lambda_j(t) - \left\{ \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t))(1 - p_i(\boldsymbol{\lambda}(t)))|g_{ij}| \right\}^{-1} \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t)) g_{ij}. \tag{13}$$

This gives rise to a gradient SM algorithm.

4.2 Using first-order Taylor approximation

First, notice that $\ln\cosh(u) = \ln \frac{\exp(u)+\exp(-u)}{2}$ for $u \in (-\infty, \infty)$ is even while $\ln\cosh\sqrt{u}$ for $u \in [0, \infty)$ is concave (Jaakkola and Jordan 1997). It is easy to obtain

$$\begin{aligned} \ln(1 + \exp(\boldsymbol{\lambda}^T \mathbf{g}_i)) &= \ln 2 + \frac{\boldsymbol{\lambda}^T \mathbf{g}_i}{2} + \ln\cosh\left(\frac{\boldsymbol{\lambda}^T \mathbf{g}_i}{2}\right) \\ &= \ln 2 + \frac{\boldsymbol{\lambda}^T \mathbf{g}_i}{2} + \ln\cosh\left(\frac{|\boldsymbol{\lambda}^T \mathbf{g}_i|}{2}\right). \end{aligned} \tag{14}$$

Let $\sqrt{u} = \frac{|\boldsymbol{\lambda}^T \mathbf{g}_i|}{2}$. Then it follows from the concavity[1] of $\ln\cosh\sqrt{u}$ that

$$\begin{aligned} \ln\cosh\left(\frac{|\boldsymbol{\lambda}^T \mathbf{g}_i|}{2}\right) &\le \ln\cosh\left(\frac{|\boldsymbol{\lambda}(t)^T \mathbf{g}_i|}{2}\right) + \beta_i(t)\left(\frac{(\boldsymbol{\lambda}^T \mathbf{g}_i)^2}{4} - \frac{(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)^2}{4}\right) \\ &= \ln\cosh\left(\frac{|\boldsymbol{\lambda}(t)^T \mathbf{g}_i|}{2}\right) + \frac{1}{4}(\boldsymbol{\lambda} - \boldsymbol{\lambda}(t))^T \beta_i(t) \mathbf{g}_i \mathbf{g}_i^T (\boldsymbol{\lambda} + \boldsymbol{\lambda}(t)), \end{aligned}$$

[1] It is well-known that $\ln\cosh\sqrt{u}$ is concave. Nevertheless, we present a new proof in Appendix 1 because the proof procedure will be useful in the sequel.

where $\beta_i(t)$ stands for the derivative of $\ln\cosh\sqrt{u}$ at $\sqrt{u} = |\boldsymbol{\lambda}(t)^T\mathbf{g}_i|/2$, and $\beta_i(t) = \frac{\tanh(|\boldsymbol{\lambda}(t)^T\mathbf{g}_i|/2)}{|\boldsymbol{\lambda}(t)^T\mathbf{g}_i|}$ when $\boldsymbol{\lambda}(t)^T\mathbf{g}_i \neq 0$ and $\beta_i(t) = \frac{1}{2}$ otherwise. Thus, we obtain a quadratic surrogate function

$$\begin{aligned} Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) &= n\ln 2 + \sum_{i=1}^{n}\left\{\frac{\boldsymbol{\lambda}^T\mathbf{g}_i}{2} + \ln\cosh\left(\frac{\boldsymbol{\lambda}(t)^T\mathbf{g}_i}{2}\right)\right\} \\ &\quad + \frac{1}{4}(\boldsymbol{\lambda} - \boldsymbol{\lambda}(t))^T\left\{\sum_{i=1}^{n}\beta_i(t)\mathbf{g}_i\mathbf{g}_i^T\right\}(\boldsymbol{\lambda} + \boldsymbol{\lambda}(t)) \\ &= L_b(\boldsymbol{\lambda}(t)) + \sum_{i=1}^{n}\frac{(\boldsymbol{\lambda} - \boldsymbol{\lambda}(t))^T\mathbf{g}_i}{2} \\ &\quad + \frac{1}{4}(\boldsymbol{\lambda} - \boldsymbol{\lambda}(t))^T\left\{\sum_{i=1}^{n}\beta_i(t)\mathbf{g}_i\mathbf{g}_i^T\right\}(\boldsymbol{\lambda} + \boldsymbol{\lambda}(t)). \end{aligned} \tag{15}$$

Minimization of $Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ w.r.t. $\boldsymbol{\lambda}$ results in a new one-step SM algorithm

$$\boldsymbol{\lambda}(t+1) = -\left\{\sum_{i=1}^{n}\beta_i(t)\mathbf{g}_i\mathbf{g}_i^T\right\}^{-1}\sum_{i=1}^{n}\mathbf{g}_i. \tag{16}$$

4.3 Using the low quadratic bound principle

The original idea of the low quadratic bound principle was proposed by (Böhning and Lindsay 1988). More specifically, let $L(\boldsymbol{\theta})$ be the objective function to be maximized, $\nabla L(\boldsymbol{\theta})$ the Fisher score vector and $\nabla^2 L(\boldsymbol{\theta})$ the Hessian matrix at $\boldsymbol{\theta} \in \mathbb{R}^q$. The low quadratic bound principle aims at finding a negative definite $q \times q$ matrix $\mathbf{B}$ such that $\nabla^2 L(\boldsymbol{\theta}) \succeq \mathbf{B}$ for all $\boldsymbol{\theta}$.[2] Thus, one can define the surrogate function $Q(\boldsymbol{\theta}|\boldsymbol{\phi})$ of $L(\boldsymbol{\theta})$ as

$$Q(\boldsymbol{\theta}|\boldsymbol{\phi}) = L(\boldsymbol{\phi}) + (\boldsymbol{\theta} - \boldsymbol{\phi})^T\nabla L(\boldsymbol{\phi}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\phi})^T\mathbf{B}(\boldsymbol{\theta} - \boldsymbol{\phi}).$$

Clearly, $L(\boldsymbol{\theta}) - Q(\boldsymbol{\theta}|\boldsymbol{\phi})$ attains its minimum at $\boldsymbol{\theta} = \boldsymbol{\phi}$. Since $Q(\boldsymbol{\theta}|\boldsymbol{\phi})$ is a quadratic function, its concavity implies that it has only one maximum. If we let $\boldsymbol{\phi}$ be the tth estimate of $\boldsymbol{\theta}$, denoted $\boldsymbol{\theta}(t)$, then maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t))$ w.r.t. $\boldsymbol{\theta}$ yields the $(t+1)$th estimate of $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \mathbf{B}^{-1}\nabla L(\boldsymbol{\theta}(t)). \tag{17}$$

Note that if $\mathbf{B}$ is singular, we use its Moore–Penrose inverse instead. Obviously, it is a special case of the SM algorithm, and, due to its origin from the low quadratic bound principle, will be referred to as the quadratic SM algorithm in the sequel.

We now apply the low quadratic bound principle to the binary logistic regression model. First, we compute the Fisher score vector and Hessian matrix as

[2] Here $\mathbf{C} \succeq \mathbf{D}$ means $\mathbf{C} - \mathbf{D}$ is positive semidefinite.

$$\begin{aligned}\nabla L_b(\boldsymbol{\lambda}) &= \sum_{i=1}^{n} p_i(\boldsymbol{\lambda})\mathbf{g}_i,\\ \nabla^2 L_b(\boldsymbol{\lambda}) &= \sum_{i=1}^{n} p_i(\boldsymbol{\lambda})(1-p_i(\boldsymbol{\lambda}))\mathbf{g}_i\mathbf{g}_i^T.\end{aligned} \tag{18}$$

This leads to the following second-order Taylor series approximation of the objective function $L_b(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda}(t)$:

$$Q_n(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = L(\boldsymbol{\lambda}(t)) + (\boldsymbol{\lambda}-\boldsymbol{\lambda}(t))^T\nabla L(\boldsymbol{\lambda}(t)) + \frac{1}{2}(\boldsymbol{\lambda}-\boldsymbol{\lambda}(t))^T\nabla^2 L(\boldsymbol{\lambda}(t))(\boldsymbol{\lambda}-\boldsymbol{\lambda}(t)). \tag{19}$$

Using the pure Newton method, the corresponding iteration formula is

$$\boldsymbol{\lambda}(t+1) = \boldsymbol{\lambda}(t) - \left\{\sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t))(1-p_i(\boldsymbol{\lambda}(t)))\mathbf{g}_i\mathbf{g}_i^T\right\}^{-1}\sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t))\mathbf{g}_i. \tag{20}$$

On the other hand, since $p_i(\boldsymbol{\lambda})(1-p_i(\boldsymbol{\lambda})) \le \frac{1}{4}$, we have

$$\nabla^2 L_b(\boldsymbol{\lambda}) \preceq \frac{1}{4}\mathbf{G}\mathbf{G}^T,$$

where $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n]$. Now, given the tth iterates $\lambda_j(t)$'s of λ_j's, we can define a surrogate function of $L_b(\boldsymbol{\lambda})$ as

$$Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = L_b(\boldsymbol{\lambda}(t)) + (\boldsymbol{\lambda}-\boldsymbol{\lambda}(t))^T\nabla L_b(\boldsymbol{\lambda}(t)) + \frac{1}{8}(\boldsymbol{\lambda}-\boldsymbol{\lambda}(t))^T\mathbf{G}\mathbf{G}^T(\boldsymbol{\lambda}-\boldsymbol{\lambda}(t)). \tag{21}$$

Then, minimization of $Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ gives rise to the $(t+1)$th iterate of $\boldsymbol{\lambda}$, as:

$$\boldsymbol{\lambda}(t+1) = \boldsymbol{\lambda}(t) - 4(\mathbf{G}\mathbf{G}^T)^{-1}\nabla L_b(\boldsymbol{\lambda}(t)). \tag{22}$$

We can see that the assumption $\sum_{j=1}^{m}|g_{ij}| \le 1$ is not necessary for this SM algorithm.

4.4 Different combinations of the basic approaches

Depending upon the problem at hand, usually one of the three approaches mentioned in Sect. 2 is used to construct a surrogate function. However, when none of these approaches can give a closed-form solution, one may consider using multiple approaches in tandem. Here we illustrate some combination approaches in the context of the binary logistic regression model. We will first consider the combination of Jensen's inequality and the first-order Taylor approximation, and will see that it works well independent of the order in which they are combined. Next, we will consider the combination of Jensen's inequality and the low quadratic bound principle.

Combination 1 We first apply Jensen's inequality to $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ in (12) and then apply first-order Taylor approximation to the $\ln(\cdot)$ function. Specifically, by

$$\ln(u) \le \ln(v) + \frac{u-v}{v} \quad \text{for } u, v > 0 \tag{23}$$

and letting $u = 1 + \exp(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)) + \boldsymbol{\lambda}(t)^T \mathbf{g}_i)$ and $v = 1 + \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)$, we have,

$$\begin{aligned}
&\ln\left[1 + \exp\left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)) + \boldsymbol{\lambda}(t)^T \mathbf{g}_i\right)\right] \\
&\quad \le \ln[1 + \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)] + \frac{(\exp(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t))) - 1)\exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)}{1 + \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)}.
\end{aligned}$$

By combining this with $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$, we obtain a new surrogate function for $L_b(\boldsymbol{\lambda})$:

$$\begin{aligned}
Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) &= \sum_{i=1}^{n} \ln\left(1 + \exp\left(\sum_{j=1}^{m} \lambda_j(t) g_{ij}\right)\right) \\
&\quad + \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t)) \sum_{j=1}^{m} |g_{ij}| \left\{\exp\left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t))\right) - 1\right\}. \qquad (24)
\end{aligned}$$

Since the partial derivative of $Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ w.r.t. λ_j is

$$\begin{aligned}
\frac{\partial Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))}{\partial \lambda_j} &= \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t)) g_{ij} \exp\left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t))\right) \\
&= \sum_{i \in S_j^+} p_i(\boldsymbol{\lambda}(t)) |g_{ij}| \exp(\lambda_j - \lambda_j(t)) \\
&\quad - \sum_{i \in S_j^-} p_i(\boldsymbol{\lambda}(t)) |g_{ij}| \exp(\lambda_j(t) - \lambda_j),
\end{aligned}$$

where $S_j^+ = \{i : g_{ij} > 0\}$ and $S_j^- = \{i : g_{ij} < 0\}$, it is easy to find an exact analytical solution of $\operatorname{argmin}_{\boldsymbol{\lambda}} Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ as

$$\lambda_j(t+1) = \lambda_j(t) + \frac{1}{2} \ln\left(\frac{\sum_{i \in S_j^-} |g_{ij}| p_i(\boldsymbol{\lambda}(t))}{\sum_{i \in S_j^+} |g_{ij}| p_i(\boldsymbol{\lambda}(t))}\right). \qquad (25)$$

Combination 2 The second combination approach first applies the first-order Taylor approximation and then Jensen's inequality. Now let $u = 1 + \exp(\sum_{j=1}^{m} \lambda_j g_{ij})$ and $v = 1 + \exp(\sum_{j=1}^{m} \lambda_j(t) g_{ij})$ in (23), we have

$$\begin{aligned}
&\ln\left(1 + \exp\left(\sum_{j=1}^{m} \lambda_j g_{ij}\right)\right) \\
&\quad \le \ln\left(1 + \exp\left(\sum_{j=1}^{m} \lambda_j(t) g_{ij}\right)\right) + \frac{\exp(\sum_{j=1}^{m} \lambda_j g_{ij}) - \exp(\sum_{j=1}^{m} \lambda_j(t) g_{ij})}{1 + \exp(\sum_{j=1}^{m} \lambda_j(t) g_{ij})}.
\end{aligned}$$

It thus follows from $L_b(\boldsymbol{\lambda})$ in (9) that

$$L_b(\boldsymbol{\lambda}) \le \sum_{i=1}^{n} \ln\left(1 + \exp\left(\sum_{j=1}^{m} \lambda_j(t) g_{ij}\right)\right)$$

$$
\begin{aligned}
&+\sum_{i=1}^{n}\frac{\exp(\sum_{j=1}^{m}\lambda_j g_{ij})-\exp(\sum_{j=1}^{m}\lambda_j(t)g_{ij})}{1+\exp(\sum_{j=1}^{m}\lambda_j(t)g_{ij})}\\
&=\sum_{i=1}^{n}\ln\left(1+\exp\left(\sum_{j=1}^{m}\lambda_j(t)g_{ij}\right)\right)\\
&\quad+\sum_{i=1}^{n}p_i(\boldsymbol{\lambda}(t))\left[\exp\left(\sum_{j=1}^{m}(\lambda_j-\lambda_j(t))g_{ij}\right)-1\right]\\
&\equiv Q_*(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)).
\end{aligned}
$$

Using Jensen's inequality, we have

$$
\begin{aligned}
\exp\left(\sum_{j=1}^{m}(\lambda_j-\lambda_j(t))g_{ij}\right)&=\exp\left(\sum_{j=1}^{m}|g_{ij}|\frac{g_{ij}}{|g_{ij}|}(\lambda_j-\lambda_j(t))+(1-\alpha_i)0\right)\\
&\le 1-\alpha_i+\sum_{j=1}^{m}|g_{ij}|\exp\left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j-\lambda_j(t))\right).
\end{aligned}
$$

Inserting this inequality into $Q_*(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$, we again obtain the surrogate function $Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ and the iterative equation given in (25).

Clearly, this is a standard SM algorithm. Note that this algorithm is equivalent to the parallel Bregman distance algorithm for binary logistic regression proposed by (Collins et al. 2002). However, our derivation is much simpler because we only utilize Jensen's inequality with the convexity of $\ln(1+\exp(u))$ and first-order Taylor approximation with the concavity of $\ln(u)$.

Combination 3 The point of departure of the third combination approach is from the surrogate function $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ defined in (12). As

$$
\begin{aligned}
\frac{\partial Q_z^2(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))}{\partial\lambda_j^2}&=\sum_{i=1}^{n}|g_{ij}|\frac{\exp(\boldsymbol{\lambda}(t)^T\mathbf{g}_i+\frac{g_{ij}}{|g_{ij}|}(\lambda_j-\lambda_j(t)))}{1+\exp(\boldsymbol{\lambda}(t)^T\mathbf{g}_i+\frac{g_{ij}}{|g_{ij}|}(\lambda_j-\lambda_j(t)))}\\
&\quad\times\left[1-\frac{\exp(\boldsymbol{\lambda}(t)^T\mathbf{g}_i+\frac{g_{ij}}{|g_{ij}|}(\lambda_j-\lambda_j(t)))}{1+\exp(\boldsymbol{\lambda}(t)^T\mathbf{g}_i+\frac{g_{ij}}{|g_{ij}|}(\lambda_j-\lambda_j(t)))}\right]\\
&\le\frac{1}{4}\sum_{i=1}^{n}|g_{ij}|,
\end{aligned}
$$

we apply the low quadratic bound principle to $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$, leading to another surrogate function

$$
Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))=L_b(\boldsymbol{\lambda}(t))+(\boldsymbol{\lambda}-\boldsymbol{\lambda}(t))^T\nabla L_b(\boldsymbol{\lambda}(t))+\frac{1}{8}(\boldsymbol{\lambda}-\boldsymbol{\lambda}(t))^T\mathbf{D}(\boldsymbol{\lambda}-\boldsymbol{\lambda}(t)), \tag{26}
$$

where $\mathbf{D} = \mathrm{diag}(\sum_{i=1}^n |g_{i1}|, \ldots, \sum_{i=1}^n |g_{im}|)$ is a diagonal matrix, and we use $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = L_b(\boldsymbol{\lambda})$ and $\nabla Q_z(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)) = \nabla L_b(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}(t)$. Thus, we have

$$\lambda_j(t+1) = \lambda_j(t) - 4\left(\sum_{i=1}^n |g_{ij}|\right)^{-1} \sum_{i=1}^n p_i(\boldsymbol{\lambda}(t))g_{ij}, \quad j = 1, \ldots, m. \tag{27}$$

Apparently, $Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ is also a surrogate function for the log-likelihood function $L_b(\boldsymbol{\lambda})$ through combining Jensen's inequality and the low quadratic bound principle.

4.5 Theoretical analysis

We can see that the surrogate function for an objective function is not unique. By using (combinations of) different approaches from Sect. 2, different surrogate functions and consequently different SM algorithms can be devised. Table 3 compares the various SM algorithms proposed in the previous subsections, and their needs for matrix inversion are shown in Table 4.

From Sect. 4.4, it can be shown that, for the same $\boldsymbol{\lambda}(t)$,

$$L_b(\boldsymbol{\lambda}) \le Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \le Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)), \tag{28}$$

$$L_b(\boldsymbol{\lambda}) \le Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \le Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)). \tag{29}$$

Again considering $\boldsymbol{\lambda}(t+1)$ in (25), we have

$$Q_z(\boldsymbol{\lambda}(t+1)|\boldsymbol{\lambda}(t)) \le Q_c(\boldsymbol{\lambda}(t+1)|\boldsymbol{\lambda}(t)) \le Q_c(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)) = L_b(\boldsymbol{\lambda}(t)) = Q_z(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)),$$
$$Q_z(\boldsymbol{\lambda}(t+1)|\boldsymbol{\lambda}(t)) \le Q_m(\boldsymbol{\lambda}(t+1)|\boldsymbol{\lambda}(t)) \le Q_m(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)) = L_b(\boldsymbol{\lambda}(t)) = Q_z(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)).$$

This implies that the iterative procedure based on either (25) or (27) defines a generalized SM algorithm w.r.t. the surrogate function $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$. Therefore, we see that a standard SM algorithm w.r.t. one surrogate function may at the same time be a generalized SM algorithm w.r.t. another surrogate function.

SM-1 is a gradient SM algorithm. Like the gradient EM algorithm, its convergence is not guaranteed. Here, since the SM-k ($k = 2, \ldots, 5$) algorithms are standard SM algorithms, we consider their convergence properties. For SM-3 and SM-5, their corresponding surrogate functions $Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ and $Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ are clearly continuous in both $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}(t)$. For SM-2,

Table 3 General comparison of the proposed SM algorithms and the pure Newton method for the binary logistic regression

Method	Surrogate function	Iterative equation	Approach(es) used
SM-1	$Q_z(\boldsymbol{\lambda}\|\boldsymbol{\lambda}(t))$ in (12)	(13)	Jensen's inequality
SM-2	$Q_f(\boldsymbol{\lambda}\|\boldsymbol{\lambda}(t))$ in (15)	(16)	First-order Taylor approximation
SM-3	$Q_q(\boldsymbol{\lambda}\|\boldsymbol{\lambda}(t))$ in (21)	(22)	Low quadratic bound principle
SM-4	$Q_c(\boldsymbol{\lambda}\|\boldsymbol{\lambda}(t))$ in (24)	(25)	Jensen's inequality + first-order Taylor approximation
SM-5	$Q_m(\boldsymbol{\lambda}\|\boldsymbol{\lambda}(t))$ in (26)	(27)	Jensen's inequality + low quadratic bound principle
Newton's	$Q_n(\boldsymbol{\lambda}\|\boldsymbol{\lambda}(t))$ in (19)	(20)	

Table 4 Comparison on the needs for matrix inversion of the proposed SM algorithms and the pure Newton method for binary logistic regression

Method	Matrix inversion?
SM-1	No need for matrix inversion
SM-2	Invert an $m \times m$ matrix at each iteration
SM-3	Invert an $m \times m$ matrix once during the whole process
SM-4	No need for matrix inversion
SM-5	No need for matrix inversion
Newton's	Invert an $m \times m$ matrix at each iteration

it is easy to see from Lemma 1 that $\beta_i(t)$ is continuous in $\boldsymbol{\lambda}(t)^T \mathbf{g}_i$ at $(-\infty, +\infty)$. As a result, we obtain that $Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ is continuous in $\boldsymbol{\lambda}(t)$, and hence $Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ is continuous in both $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}(t)$. As for SM-4, we choose to regard it as a generalized SM algorithm w.r.t. the surrogate function $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ in (12), which is continuous in both $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}(t)$. On the other hand, from (19), we have $\nabla^2 L_b(\boldsymbol{\lambda}) \succeq 0$. Consequently, $L_b(\boldsymbol{\lambda})$ is convex. We again note that $L_b(\boldsymbol{\lambda})$ is bounded below (≥ 0). This shows that $L_b(\boldsymbol{\lambda})$ only has a unique stationary point which is the local minimum. By Theorem 2, we thus have the following corollary.

Corollary 1 *The limit point of any sequence* $\{\boldsymbol{\lambda}(t)\}$ *of one of the SM-k* $(k = 2, \ldots, 5)$ *algorithms is the global minimum of* $L_b(\boldsymbol{\lambda})$ *and* $L_b(\boldsymbol{\lambda}(t))$ *converges monotonically to* $L_b(\boldsymbol{\lambda}^*)$ *for the global minimum* $\boldsymbol{\lambda}^*$.

With a variety of different possibilities, a natural question to ask is what criteria should be used to guide the design of a good surrogate function. Intuitively, one criterion that could be used is the closeness of a surrogate function to the original objective function. For example, the closer is the surrogate function to the objective function, the better it will be. Another possible criterion is the tractability of the M-step. For example, a closed-form update equation is more desirable. In other words, we want the surrogate function to be both *efficient* and *effective*. In practice, however, there has to be a tradeoff between these two criteria.

Now, we discuss this issue by taking our proposed SM algorithms as concrete examples. Specifically, we have that, for the same $\boldsymbol{\lambda}(t)$,

$$Q_n(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \quad (\text{or } L_b(\boldsymbol{\lambda})) \leq Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \leq Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \leq Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)). \tag{30}$$

The proof can be found in Appendix 2. Since their corresponding SM algorithms are standard, we can order the convergence rate of these algorithms as

$$\text{the pure Newton method} \geq \text{SM-2} \geq \text{SM-3} \geq \text{SM-5}. \tag{31}$$

This shows that the closer is a surrogate function to the objective function, the faster the rate of convergence of the standard SM algorithm corresponding this surrogate function will be. On the other hand, from (16) and (22), we can see that both SM-2 and SM-3 based on $Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ and $Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ amount essentially to minimizing $L_b(\boldsymbol{\lambda})$ by the pure Newton method, but with the Hessian matrix $\nabla^2 L_b(\boldsymbol{\lambda})$ replaced by an approximated matrix. They can avoid the non-convergent problem of the pure Newton method. SM-2 has the same computational cost as Newton's method. Since SM-3 uses a constant matrix (i.e., $\mathbf{B}$), it only needs to compute the inverse of this constant matrix once during the whole iterative process. However, SM-5 does not need to invert any matrix. Thus, in general, there may be a tradeoff between the two criteria.

Further, going back to (28) and (29), we have that the surrogate function $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ is superior to $Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ and $Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$. However, while SM-1 based on $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ does not have a closed-form solution for the M-step, it is easy to show that an exact analytical solution exists for SM-4 or SM-5 based on $Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ or $Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$. It is worth noting that although we have (28), SM-1 $\geq$ SM-4 does not always hold because SM-1 is a gradient algorithm. Given the same $\boldsymbol{\lambda}(t)$, we denote its next estimates by $\boldsymbol{\lambda}^{(1)}(t+1)$ and $\boldsymbol{\lambda}^{(4)}(t+1)$ from SM-1 and SM-4, respectively. For SM-1, $\boldsymbol{\lambda}^{(1)}(t+1)$ may not be the minimum of $Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$. Consequently, we do not ensure that

$$Q_z(\boldsymbol{\lambda}^{(1)}(t+1)|\boldsymbol{\lambda}(t)) \leq Q_c(\boldsymbol{\lambda}^{(4)}(t+1)|\boldsymbol{\lambda}(t)).$$

In other words, we are not able to guarantee that SM-1 is faster than SM-4. However, for SM-1 and SM-5, it can be shown from the last paragraph in Sect. 4.4 that

$$\nabla^2 Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \leq \frac{1}{4}\mathbf{D}.$$

Thus the Rayleigh quotient of $\nabla^2 Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ is smaller than that of $\frac{1}{4}\mathbf{D}$. Therefore,

$$\eta = \|(\nabla^2 Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)))^{-1}\nabla^2 L(\boldsymbol{\lambda})\| \geq \|4\mathbf{D}^{-1}\nabla^2 L(\boldsymbol{\lambda})\|.$$

This implies that the dominant eigenvalue of $\mathbf{I} - (\nabla^2 Q_z(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)))^{-1}\nabla^2 L(\boldsymbol{\lambda})$ is not smaller than that of $\mathbf{I} - 4\mathbf{D}^{-1}\nabla^2 L(\boldsymbol{\lambda})$. As described in the so-called Ostrowski's theorem (Ostrowski 1960, Chap. 18), the dominant eigenvalue determines the convergence rate of the corresponding algorithm. Thus SM-1 is faster than SM-5, i.e., we still have

$$\text{SM-1} \geq \text{SM-5}. \tag{32}$$

4.6 Experimental analysis

In this subsection we empirically evaluate the SM algorithms summarized in Table 3 for the binary logistic regression model. Our goal is to further validate the theoretical analysis given in Sect. 4.5 from an experimental perspective. Specifically, we attempt to achieve the following purposes:

(a) Illustrate the tradeoff between efficiency and effectiveness;
(b) Illustrate the tradeoff between training and testing.

In our experiments we use the pure Newton method for baseline comparison due to its relationship with the SM algorithms given in (30) and (31). An empirical comparison of some SM algorithms with other numerical methods such as conjugate gradient and quasi-Newton have been systematically studied in (Minka 2003). In the experiments, we use two synthetic data sets similar to those used in (Collins et al. 2002) and two real-world data sets. The code is implemented in MATLAB and it is available from the homepage of the first author, and the experiments are run on a Pentium 2.79 GHz PC with 2.00 GB RAM. We use the same initial values of the λ_{ij} to implement these six algorithms. Specifically, we use two initialization methods: one is to randomly generate $\lambda_{ij}(0)$ from a uniform distribution over $[-1, 1]$, i.e., $\lambda_{ij}(0) \sim U([-1, 1])$, and another is to set $\lambda_{ij}(0) = 0$. From (7), the latter method implies that $\hat{p}(y_i = 1|\mathbf{x}_i, \boldsymbol{\lambda}(0)) = \hat{p}(y_i = -1|\mathbf{x}_i, \boldsymbol{\lambda}(0)) = 1/2$ for $i = 1, \ldots, n$. For all four datasets, we run all the six algorithms until $|L_b(\boldsymbol{\lambda}(t+1)) - L_b(\boldsymbol{\lambda}(t))|/L_b(\boldsymbol{\lambda}(0)) < 0.00001$.

Simulated data The first data set consists of 3000 data points $\mathbf{x}_i \in \mathbb{R}^{100}$ sampled randomly from the normal distribution with zero mean and identity covariance matrix. To label these points, we first randomly generate a 100-dimensional hyperplane represented by a vector $\mathbf{w} \in \mathbb{R}^{100}$ subject to $\|\mathbf{w}\| = 1$ and then assign the label $y_i = \text{sgn}(\mathbf{w}^T \mathbf{x}_i)$ to each $\mathbf{x}_i$. After this labeling step, we perturb each point $\mathbf{x}_i$ by adding a random noise term $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, 0.2\mathbf{I})$, leading to a new noisy data point $\mathbf{z}_i$. We use 1000 points for training and the remaining 2000 points for testing. We run our experiments using two data sets, i.e., $\{\mathbf{x}_i\}$ without noise and $\{\mathbf{z}_i\}$ with noise. Specifically, we set $h_j(\mathbf{x}_i) = x_{ij}$ and $h_j(\mathbf{z}_i) = z_{ij}$, respectively, for the two data sets. In this case, we have $n = 1000$ and $m = 100$. For $i = 1, \dots, n$ and $j = 1, \dots, m$, we calculate $g_{ij} = -y_i h_j(\mathbf{x}_i)$ (or $g_{ij} = -y_i h_j(\mathbf{z}_i)$) and set $g_{ij} = \frac{g_{ij}}{\sum_{j=1}^{100} |g_{ij}|}$ such that $\sum_{j=1}^{100} |g_{ij}| \leq 1$.

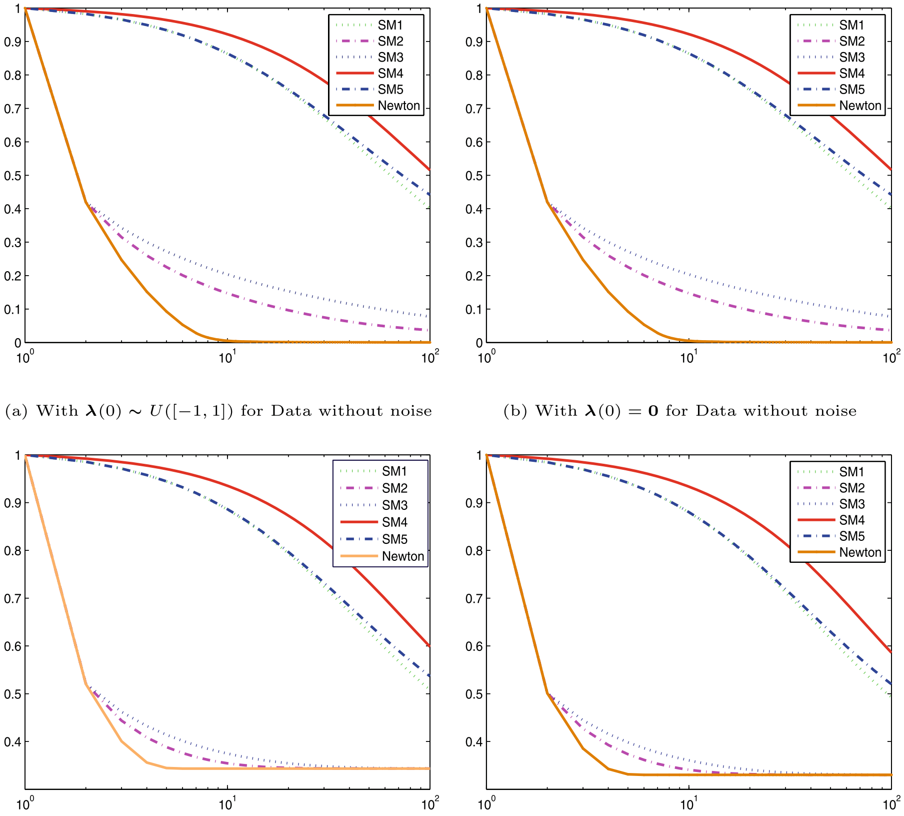
Text data We also evaluate the SM algorithms on two text categorization tasks using the WebKB (Craven et al. 1998) and NewsGroup (Joachims 1997) data sets. The WebKB data set contains web pages gathered from computer science departments in several universities. The pages can be divided into seven categories. Here we run the binary logistic regression model on the classes faculty and course, with a total of 2054 pages. The NewsGroup data set consists of 20 classes. We use the classes alt.atheism and comp.graphics, with a total of 1985 words. Based on the information gain criterion, 300 features (i.e., $m = 300$) are selected for WebKB and 1000 features (i.e., $m = 1000$) for NewsGroup. We then define a feature as

$$h_j(\mathbf{x}_k) = \frac{N_j(\mathbf{x}_k)}{N(\mathbf{x}_k)},$$

where $N_j(\mathbf{x}_k)$ is the number of occurrences of feature j in document $\mathbf{x}_k$ and $N(\mathbf{x}_k)$ is the total number of occurrences of all features in document $\mathbf{x}_k$. In the experiments, we specify 1398 training samples and 656 test samples for WebKB dataset, and 1390 training samples and 595 test samples for NewsGroup dataset.

Figures 1 and 2 show the training loss, with values normalized to 1, for the two initialization methods on these four data sets. Note that the x-axis is in log scale for all plots. Moreover, to facilitate comparison and visualization, we illustrate the training losses of the first 100 iterations, although some of the algorithms have converged and others have not converged before 100 iterations. As we can see, all six algorithms are not sensitive to the initial values of the λ_{ij} and converge though with different rates. Obviously, the convergence of SM-2, SM-3, SM-4 and SM-5 follow from the basic properties of SM algorithms. For SM-1 and the pure Newton method, they also converge in our experiments. However, as is well known, the convergence of SM-1 and the pure Newton method is generally not guaranteed. The orderings of different methods in terms of their convergence rate are same as those in (31) and (32). In addition, SM-1 $\geq$ SM-4 holds for the two simulated data sets, while it does not hold for the two text data sets. This is in full agreement with our theoretical analysis in Sect. 4.5.

Since the performance of the algorithms is almost the same for different initial values of the λ_{ij}, the experimental results, reported in Tables 5 and 6, are based on initial values of the λ_{ij} chosen randomly from $U([-1, 1])$. For SM-2 and the pure Newton method, we need to invert an $m \times m$ matrix at each iteration (see Table 4). Although SM-2 and the pure Newton method take very few iterations to converge, they become very inefficient for larger values of m due to the need for large storage. For SM-3, we need to invert an $m \times m$ matrix only once for all the iterations. Hence its computational cost is lower. For the other methods, their computational costs are even lower. These can be seen from Table 5, in



(a) With $\boldsymbol{\lambda}(0) \sim U([-1,1])$ for Data without noise

(b) With $\boldsymbol{\lambda}(0) = \mathbf{0}$ for Data without noise

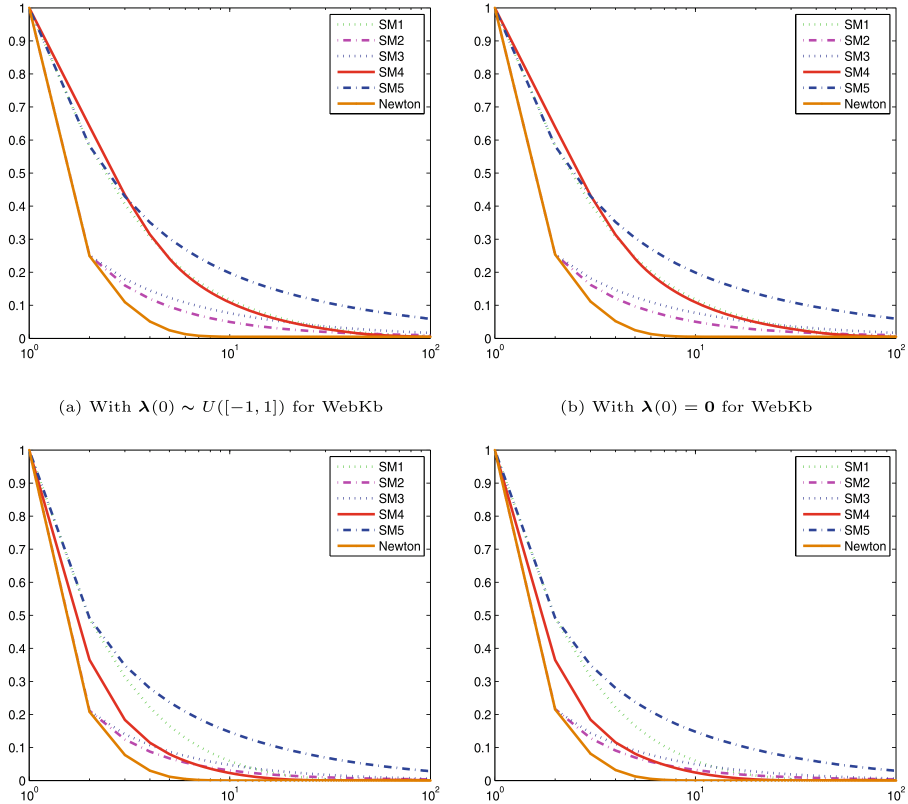
(c) With $\boldsymbol{\lambda}(0) \sim U([-1,1])$ for Data with noise

(d) With $\boldsymbol{\lambda}(0) = \mathbf{0}$ for Data with noise

Fig. 1 Training loss vs. number of iterations

which the bottom of each table entry gives the corresponding number of iterations required before convergence. Thus, there exists a trade-off between the convergence rate and the computational cost. SM-2 and the pure Newton method are inefficient for high-dimensional data, although their convergence rates are the fastest.

We also report the classification accuracies on the test data in Table 6. On the simulated data without noise, the pure Newton method, SM-2 and SM-3 outperform SM-1, SM-4 and SM-5. This shows that the classification accuracy is consistent with the convergence rate for noiseless datasets. However, on the noisy simulated data and the two text data sets, the classification results are different from those for the simulated data sets. Specifically, the classification accuracies of SM-2, SM-3 and the pure Newton method slightly decrease. In contrast, SM-1, SM-4 and SM-5 are rather robust to noise, and they now give higher accuracy than SM-2, SM-3 and the pure Newton method. Moreover, SM-5 gives the best classification performance although it is the worst in terms of convergence rate. Thus, an algorithm with higher convergence rate does not always have higher classification accuracy. Since most real-world datasets are noisy in nature, we think that SM-4 and SM-5 are the best choices when considering both computational cost and classification accuracy.



(a) With $\boldsymbol{\lambda}(0) \sim U([-1,1])$ for WebKb

(b) With $\boldsymbol{\lambda}(0) = \mathbf{0}$ for WebKb

(c) With $\boldsymbol{\lambda}(0) \sim U([-1,1])$ for NewsGroup

(d) With $\boldsymbol{\lambda}(0) = \mathbf{0}$ for NewsGroup

Fig. 2 Training loss vs. number of iterations

5 CSM and SCM algorithms for binary logistic regression model

Now we consider applying the CSM and SCM algorithms to the logistic regression model. The loss function $L(\boldsymbol{\lambda})$ can be regarded as a function $L_b(\lambda_1, \ldots, \lambda_m)$ of multiple variables λ_i's. First, if we employ the parallel-update scheme, it is easy to see that the standard, generalized and gradient SM algorithms given in the previous sections can also be regarded as CSM or SCM algorithms. On the other hand, if we employ the sequential-update scheme, it is easy to obtain a CSM or SCM algorithm from one of these SM algorithms by replacing $p_i(\boldsymbol{\lambda}(t))$ with

$$p_i(\mathbf{w}) = \frac{\exp(\mathbf{w}^T \mathbf{g}_i)}{1 + \exp(\mathbf{w}^T \mathbf{g}_i)},$$

where $\mathbf{w} = (\lambda_1(t), \ldots, \lambda_i(t), \lambda_{i+1}(t+1), \ldots, \lambda_m(t+1))^T$.

Now, we consider in more detail the application of CSM algorithms to an extension of the logistic regression model. We change $\hat{p}(y_i|\mathbf{x}_i, \boldsymbol{\lambda})$ in (7) to

$$\hat{p}(y_i|\mathbf{x}_i, \boldsymbol{\lambda}, b) = \frac{1}{1 + \exp(\boldsymbol{\lambda}^T \mathbf{g}_i + b)}, \tag{33}$$

Table 5 Total CPU time (in seconds) / number of iterations required until convergence

Dataset	SM-1	SM-2	SM-3	SM-4	SM-5	Newton's
Without noise	14.5781	21.8750	0.0156[a] +2.5469	0.0469[b] +36.7344	8.3594	3.2656
(# of iterations)	3511	256	1287	3842	4105	42
With noise	3.8906	3.4531	0.0156[a] +0.1875	0.0313[b] +13.2344	2.9375	0.5938
(# of iterations)	943	40	86	1359	1507	8
WebKB	2.1875	98.1406	0.0938[a] +2.7500	0.0469[b] +1.9531	5.2656	6.7813
(# of iterations)	147	191	357	119	957	15
NewsGroup	3.0313	523.4531	1.4219[a] +10.2969	0.1563[b] +2.3594	9.1719	38.5938
(# of iterations)	66	164	254	42	585	14

[a] CPU time required for inverting **B**

[b] CPU time required for finding S_j^+ and S_j^-

Table 6 Classification accuracy (%) after convergence

Dataset	SM-1	SM-2	SM-3	SM-4	SM-5	Newton's
Without noise	94.55	94.80	**95.10**	94.60	94.05	94.95
With noise	82.70	82.50	82.50	82.70	**82.95**	82.60
WebKB	96.95	96.49	95.88	97.10	**97.71**	96.80
NewsGroup	94.12	93.45	85.21	94.29	**96.13**	91.76

where b is a bias term. Let us denote the corresponding loss function by $L_b(\boldsymbol{\lambda}, b)$. Let $\mathbf{g}_i^+ = (g_{i1}, \ldots, g_{im}, 1)^T$ and $\boldsymbol{\lambda}^+ = (\lambda_1, \ldots, \lambda_m, b)^T$ be the extensions of $\mathbf{g}_i$ and $\boldsymbol{\lambda}$, respectively. Note that condition (10) is no longer satisfied. However, the SM-2 and SM-3 algorithms given in the previous sections can still work because (10) is not a necessary condition for them. To use the SM-1 and SM-4 algorithms, we can simply modify $\mathbf{g}_i^+ \leftarrow \frac{1}{2}\mathbf{g}_i^+$.

We now devise a CSM algorithm that alternately updates b and $\boldsymbol{\lambda}$. First, given $b(t)$, we use $Q_1(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t), b(t))$ in the same way as $Q_c(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ in (24) for a surrogate function and then obtain $\boldsymbol{\lambda}(t+1)$ with an iterative equation as in (25). However, here we replace $p_i(\boldsymbol{\lambda}(t)) = \frac{\exp(\boldsymbol{\lambda}^T(t)\mathbf{g}_i)}{1+\exp(\boldsymbol{\lambda}^T(t)\mathbf{g}_i)}$ with $p_i(\boldsymbol{\lambda}(t), b(t)) = \frac{\exp(\boldsymbol{\lambda}^T(t)\mathbf{g}_i+b(t))}{1+\exp(\boldsymbol{\lambda}^T(t)\mathbf{g}_i+b(t))}$. Then, given $\boldsymbol{\lambda}(t+1)$, we define a surrogate function $Q_2(b|b(t), \boldsymbol{\lambda}(t+1))$ of $L_b(\boldsymbol{\lambda}(t+1), b)$ as

$$L_b(\boldsymbol{\lambda}(t+1), b(t)) + \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t+1), b(t))(e^{b-b(t)} - 1),$$

where we have used the convexity of $-\ln(\cdot)$, and then obtain $b(t+1)$ via

$$b(t+1) = b(t) + \ln \sum_{i=1}^{n} p_i(\boldsymbol{\lambda}(t+1), b(t)).$$

It is easy to see that

$$\begin{aligned} L_b(\boldsymbol{\lambda}(t+1), b(t+1)) &\le Q_2(b(t+1)|b(t), \boldsymbol{\lambda}(t+1)) \le Q_2(b(t)|b(t), \boldsymbol{\lambda}(t+1)) \\ &= L_b(\boldsymbol{\lambda}(t+1), b(t)) \le Q_1(\boldsymbol{\lambda}(t+1)|\boldsymbol{\lambda}(t), b(t)) \le Q_1(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t), b(t)) \end{aligned}$$

$$= L_b(\boldsymbol{\lambda}(t), b(t)),$$

and $Q_1(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t), b(t))$ is continuous in both $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}(t)$ while $Q_2(b|b(t), \boldsymbol{\lambda}(t+1))$ is continuous in both b and $b(t)$. Thus this CSM algorithm is also guaranteed to converge in terms of Zangwill's theorem (see (De Leeuw 1994) for more details).

6 SM algorithm for multi-class logistic regression model

In a multi-class classification problem, the response variable y_i takes value from a finite set of labels, say $\mathcal{Y} = \{1, 2, \ldots, c\}$. Each feature is a mapping $h_j : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. In the logistic regression model (LogitBoost) (Collins et al. 2002; Friedman et al. 2000), we use the following probabilistic model:

$$\hat{p}(y_i|\mathbf{x}_i, \boldsymbol{\lambda}) = \frac{\exp(\sum_{j=1}^m \lambda_j h_j(\mathbf{x}_i, y_i))}{\sum_{l\in\mathcal{Y}} \exp(\sum_{j=1}^m \lambda_j h_j(\mathbf{x}_i, l))}$$
$$= \frac{1}{\sum_{l\in\mathcal{Y}} \exp(\sum_{j=1}^m \lambda_j (h_j(\mathbf{x}_i, l) - h_j(\mathbf{x}_i, y_i)))}. \tag{34}$$

Given a training set $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, the logistic regression problem can be transformed into maximizing the conditional log-likelihood

$$L_m(\boldsymbol{\lambda}) = \sum_{i=1}^n \sum_{j=1}^m \lambda_j h_j(\mathbf{x}_i, y_i) - \sum_{i=1}^n \ln \sum_{l\in\mathcal{Y}} \exp\left(\sum_{j=1}^m \lambda_j h_j(\mathbf{x}_i, l)\right),$$

or, equivalently, into minimizing the loss

$$\tilde{L}_m(\boldsymbol{\lambda}) = \sum_{i=1}^n \ln\left[\sum_{l\in\mathcal{Y}} \exp\left(\sum_{j=1}^m \lambda_j (h_j(\mathbf{x}_i, l) - h_j(\mathbf{x}_i, y_i))\right)\right].$$

We first work on $L_m(\boldsymbol{\lambda})$ to devise a quadratic SM algorithm. Since

$$\frac{\partial L_m(\boldsymbol{\lambda})}{\partial \lambda_s} = \sum_{i=1}^n \left[h_s(\mathbf{x}_i, y_i) - \sum_{l\in\mathcal{Y}} \hat{p}(l|\mathbf{x}_i, \boldsymbol{\lambda}) h_s(\mathbf{x}_i, l)\right],$$

$$\frac{\partial^2 L_m(\boldsymbol{\lambda})}{\partial \lambda_s \partial \lambda_r} = -\sum_{i=1}^n \sum_{l\in\mathcal{Y}} \hat{p}(l|\mathbf{x}_i, \boldsymbol{\lambda}) h_s(\mathbf{x}_i, l)\left[h_r(\mathbf{x}_i, y_i) - \sum_{k\in\mathcal{Y}} \hat{p}(k|\mathbf{x}_i, \boldsymbol{\lambda}) h_r(\mathbf{x}_i, k)\right],$$

then

$$\nabla L_m(\boldsymbol{\lambda}) = \sum_{i=1}^n \mathbf{H}_i(\mathbf{e}_i - \mathbf{q}_i),$$

$$\nabla^2 L_m(\boldsymbol{\lambda}) = -\sum_{i=1}^n \mathbf{H}_i(\boldsymbol{\Delta}_i - \mathbf{q}_i\mathbf{q}_i^T)\mathbf{H}_i^T,$$

where $\mathbf{e}_i$ is a $c \times 1$ vector with the kth element being 1 if $y_i = k$ and 0 otherwise,

$$\mathbf{H}_i = \begin{bmatrix} h_1(\mathbf{x}_i, 1) & h_1(\mathbf{x}_i, 2) & \dots & h_1(\mathbf{x}_i, c) \\ h_2(\mathbf{x}_i, 1) & h_2(\mathbf{x}_i, 2) & \dots & h_2(\mathbf{x}_i, c) \\ \vdots & \vdots & \ddots & \vdots \\ h_m(\mathbf{x}_i, 1) & h_m(\mathbf{x}_i, 2) & \dots & h_m(\mathbf{x}_i, c) \end{bmatrix}, \qquad \mathbf{q}_i(\boldsymbol{\lambda}) = \begin{bmatrix} \hat{p}(1|\mathbf{x}_i, \boldsymbol{\lambda}) \\ \hat{p}(2|\mathbf{x}_i, \boldsymbol{\lambda}) \\ \vdots \\ \hat{p}(c|\mathbf{x}_i, \boldsymbol{\lambda}) \end{bmatrix},$$

and $\boldsymbol{\Delta}_i(\boldsymbol{\lambda}) = \mathrm{diag}(\hat{p}(1|\mathbf{x}_i, \boldsymbol{\lambda}), \hat{p}(2|\mathbf{x}_i, \boldsymbol{\lambda}), \dots, \hat{p}(c|\mathbf{x}_i, \boldsymbol{\lambda}))$.

Using the following inequality (Böhning and Lindsay 1988)

$$\boldsymbol{\Delta}_i - \mathbf{q}_i\mathbf{q}_i^T \preceq \frac{1}{2}\left[\mathbf{I} - \frac{1}{c}\mathbf{1}\mathbf{1}^T\right],$$

where $\mathbf{1}$ is the $c \times 1$ matrix of ones, we obtain

$$\nabla^2 L_m(\boldsymbol{\lambda}) \succeq -\frac{1}{2}\sum_{i=1}^{n} \mathbf{H}_i\left[\mathbf{I} - \frac{1}{c}\mathbf{1}\mathbf{1}^T\right]\mathbf{H}_i^T \triangleq \mathbf{B}.$$

Thus, we have an iterative procedure for solving $\boldsymbol{\lambda}$, as

$$\boldsymbol{\lambda}(t+1) = \boldsymbol{\lambda}(t) + \mathbf{B}^{-1}\sum_{i=1}^{n} \mathbf{H}_i(\mathbf{e}_i - \mathbf{q}_i(\boldsymbol{\lambda}(t))). \tag{35}$$

Next, we seek to derive the parallel Bregman distance algorithm for multi-class logistic regression proposed by (Collins et al. 2002) from the perspective of an SM algorithm. We work on $\tilde{L}_m(\boldsymbol{\lambda})$ and combine the first-order Taylor approximation with Jensen's inequality. First, using the concavity of $\ln(\cdot)$, we have

$$\begin{aligned} \tilde{L}_m(\boldsymbol{\lambda}) &\le \sum_{i=1}^{n} \ln\left[\sum_{l\in\mathcal{Y}} e^{\sum_{j=1}^{m}\lambda_j(t)g_{ilj}}\right] + \sum_{i=1}^{n} \frac{\sum_{l\in\mathcal{Y}} e^{\sum_{j=1}^{m}\lambda_j g_{ilj}} - \sum_{l\in\mathcal{Y}} e^{\sum_{j=1}^{m}\lambda_j(t)g_{ilj}}}{\sum_{l\in\mathcal{Y}} e^{\sum_{j=1}^{m}\lambda_j(t)g_{ilj}}} \\ &= \sum_{i=1}^{n} \ln\left[\sum_{l\in\mathcal{Y}} e^{\sum_{j=1}^{m}\lambda_j(t)g_{ilj}}\right] + \sum_{i=1}^{n}\sum_{l\in\mathcal{Y}} p(l|\mathbf{x}_i, \boldsymbol{\lambda}(t))e^{\sum_{j=1}^{m}(\lambda_j - \lambda_j(t))g_{ilj}} - n, \end{aligned}$$

where $g_{ilj} = h_j(\mathbf{x}_i, l) - h_j(\mathbf{x}_i, y_i)$ and $p(l|\mathbf{x}_i, \boldsymbol{\lambda}(t)) = \frac{\exp(\sum_{j=1}^{m}\lambda_j(t)g_{ilj})}{\sum_{l\in\mathcal{Y}}\exp(\sum_{j=1}^{m}\lambda_j(t)g_{ilj})}$. For any i and l, we assume that $\sum_{j=1}^{m}|g_{ilj}| \le 1$. Furthermore, without loss of generality, we assume that $g_{ilj} \neq 0$ for arbitrary i, l and j. Since $\exp(\cdot)$ is convex, we have

$$\begin{aligned} \exp\left(\sum_{j=1}^{m}(\lambda_j - \lambda_j(t))g_{ilj}\right) &= \exp\left(\sum_{j=1}^{m}\frac{|g_{ilj}|g_{ilj}}{|g_{ilj}|}(\lambda_j - \lambda_j(t)) + \left(1 - \sum_{j=1}^{m}|g_{ilj}|\right)0\right) \\ &\le 1 - \sum_{j=1}^{m}|g_{ilj}| + \sum_{j=1}^{m}|g_{ilj}|\exp\left(\frac{g_{ilj}}{|g_{ilj}|}(\lambda_j - \lambda_j(t))\right). \end{aligned}$$

Thus, we obtain a surrogate function for $\tilde{L}_m(\boldsymbol{\lambda})$:

$$\tilde{Q}_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = \sum_{i=1}^{n}\ln\left[\sum_{l\in\mathcal{Y}}\exp\left(\sum_{j=1}^{m}\lambda_j(t)g_{ilj}\right)\right] - \sum_{i=1}^{n}\sum_{l\in\mathcal{Y}} p(l|\mathbf{x}_i, \boldsymbol{\lambda}(t))\sum_{j=1}^{m}|g_{ilj}|$$

$$+\sum_{i=1}^{n}\sum_{l\in\mathcal{Y}} p(l|\mathbf{x}_i,\boldsymbol{\lambda}(t))\sum_{j=1}^{m}|g_{ilj}|\exp\left(\frac{g_{ilj}}{|g_{ilj}|}(\lambda_j-\lambda_j(t))\right). \qquad (36)$$

We are interested in the minimization of $\tilde{Q}_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ w.r.t. $\boldsymbol{\lambda}$. Taking the derivatives of $\tilde{Q}_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ w.r.t. $\boldsymbol{\lambda}$:

$$\begin{aligned}\frac{\partial \tilde{Q}_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))}{\partial\lambda_s} &= \sum_{i=1}^{n}\sum_{l\in\mathcal{Y}} p(l|\mathbf{x}_i,\boldsymbol{\lambda}(t))\, g_{ils}\exp\left(\frac{g_{ils}}{|g_{ils}|}(\lambda_s-\lambda_s(t))\right)\\ &= \sum_{(i,l)\in S_s^+} p(l|\mathbf{x}_i,\boldsymbol{\lambda}(t))|g_{ils}|\exp(\lambda_s-\lambda_s(t))\\ &\quad - \sum_{(i,l)\in S_s^-} p(l|\mathbf{x}_i,\boldsymbol{\lambda}(t))|g_{ils}|\exp(\lambda_s(t)-\lambda_s),\end{aligned}$$

where $S_s^+=\{(i,l): g_{ils}>0\}$ and $S_s^-=\{(i,l): g_{ils}<0\}$. So the solution of $\frac{\partial\tilde{Q}_m(\lambda|\lambda(t))}{\partial\lambda_s}=0$ leads us to the $(t+1)$th estimate of λ_s, as

$$\lambda_s(t+1)=\lambda_s(t)+\frac{1}{2}\ln\left(\frac{\sum_{(i,l)\in S_s^-}|g_{ils}|\,p(l|\mathbf{x}_i,\boldsymbol{\lambda}(t))}{\sum_{(i,l)\in S_s^+}|g_{ils}|\,p(l|\mathbf{x}_i,\boldsymbol{\lambda}(t))}\right). \qquad (37)$$

Obviously,

$$\tilde{L}_m(\boldsymbol{\lambda}(t+1))\le\tilde{Q}_m(\boldsymbol{\lambda}(t+1)|\boldsymbol{\lambda}(t))\le\tilde{Q}_m(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t))=\tilde{L}_m(\boldsymbol{\lambda}(t)).$$

This is thus an SM algorithm, which is equivalent to the parallel Bregman distance algorithm of (Collins et al. 2002) for the multi-class logistic regression. It is clear that $\tilde{Q}_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ is continuous in both $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}(t)$. In addition, Lemma 2 shows that $\boldsymbol{\Delta}_i-\mathbf{q}_i\mathbf{q}_i^T\succeq 0$. Thus $\nabla^2\tilde{L}_m(\boldsymbol{\lambda})=\sum_{i=1}^n\mathbf{H}_i(\boldsymbol{\Delta}_i-\mathbf{q}_i\mathbf{q}_i^T)\mathbf{H}_i^T$ is positive semidefinite. Similar to Corollary 1, we have the following corollary.

Corollary 2 *The limit point of any sequence* $\{\boldsymbol{\lambda}(t)\}$ *of the SM algorithm defined in* (37) *is the global minimum of* $\tilde{L}_m(\boldsymbol{\lambda})$ *and* $\tilde{L}_m(\boldsymbol{\lambda}(t))$ *converges monotonically to* $\tilde{L}_m(\boldsymbol{\lambda}^*)$ *for the global minimum* $\boldsymbol{\lambda}^*$.

7 SM algorithm for AdaBoost

In this section we present SM algorithms for binary and multi-class AdaBoost. There exists a connection between AdaBoost and maximum likelihood for exponential models (Friedman et al. 2000; Lebanon and Lafferty 2001). Unlike the binary logistic regression model which is based on the minimization of (9), binary AdaBoost is based on the minimization of the exponential loss function

$$L_a(\boldsymbol{\lambda})=\sum_{i=1}^{n}\exp\left[-y_i\sum_{j=1}^{m}\lambda_j h_j(\mathbf{x}_i)\right]=\sum_{i=1}^{n}\exp\left(\sum_{j=1}^{m}\lambda_j g_{ij}\right). \qquad (38)$$

Let us denote the tth iteration of λ_j by $\lambda_j(t)$. From (38), we have

$$L_a(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \exp\left(\sum_{j=1}^{m} |g_{ij}| \frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)) + \boldsymbol{\lambda}(t)^T \mathbf{g}_i\right)$$
$$= \sum_{i=1}^{n} \exp\left(\sum_{j=1}^{m} |g_{ij}| \frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t)) + (1-\alpha_i)0\right) \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i).$$

Since $\exp(\cdot)$ is convex, it can be shown that

$$L_a(\boldsymbol{\lambda}) \le \sum_{i=1}^{n} \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i) \left\{1 - \alpha_i + \sum_{j=1}^{m} |g_{ij}| \exp\left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t))\right)\right\} \equiv Q_a(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)).$$

Clearly, $Q_a(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)) = L_a(\boldsymbol{\lambda}(t))$, and thus the right-hand side can be used as a surrogate function of $L_a(\lambda)$. Note also that $Q_a(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ has decoupled the relationship among the λ_j's. To minimize $Q_a(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ w.r.t. λ_j's, we set

$$\frac{\partial Q_a(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))}{\partial \lambda_j} = \sum_{i=1}^{n} g_{ij} \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i) \exp\left(\frac{g_{ij}}{|g_{ij}|}(\lambda_j - \lambda_j(t))\right)$$

to zero, and obtain

$$\sum_{i \in S_j^+} |g_{ij}| \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i) \exp(\lambda_j - \lambda_j(t)) = \sum_{i \in S_j^-} |g_{ij}| \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i) \exp(\lambda_j(t) - \lambda_j),$$

where $S_j^+ = \{i : g_{ij} > 0\}$ and $S_j^- = \{i : g_{ij} < 0\}$. We take log on both sides and, upon simplification, obtain the following update equation for λ_j:

$$\lambda_j(t+1) = \lambda_j(t) + \frac{1}{2} \ln\left(\frac{\sum_{i \in S_j^-} |g_{ij}| \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)}{\sum_{i \in S_j^+} |g_{ij}| \exp(\boldsymbol{\lambda}(t)^T \mathbf{g}_i)}\right).$$

As $L_a(\boldsymbol{\lambda}(t+1)) \le Q_a(\boldsymbol{\lambda}(t+1)|\boldsymbol{\lambda}(t)) \le Q_a(\boldsymbol{\lambda}(t)|\boldsymbol{\lambda}(t)) = L_a(\boldsymbol{\lambda}(t))$, local convergence is guaranteed. Notice that the derivation of our SM algorithm is equivalent to the one by Lebanon and Lafferty (2001).

There are two popular versions of multi-class AdaBoost. The first one is AdaBoost.M2 (Freund and Schapire 1997), which is based on the loss function

$$L_{m2}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{l \in \mathcal{Y}} \exp\left[\sum_{j=1}^{m} \lambda_j (h_j(\mathbf{x}_i, l) - h_j(\mathbf{x}_i, y_i))\right], \tag{39}$$

and the other is AdaBoost.MH (Schapire and Singer 1999), which is based on the loss function

$$L_{mh}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{l \in \mathcal{Y}} \exp\left[-\tilde{y}_{i,l} \sum_{j=1}^{m} \lambda_j h_j(\mathbf{x}_i, l)\right], \tag{40}$$

where

$$\tilde{y}_{i,l} = \begin{cases} +1 & \text{if } l = y_i, \\ -1 & \text{if } l \ne y_i. \end{cases}$$

Let $g_{ilj} = h_j(\mathbf{x}_i, l) - h_j(\mathbf{x}_i, y_i)$ or $g_{ilj} = -\tilde{y}_{i,l} h_j(\mathbf{x}_i, l)$, and use Jensen's inequality with $\sum_{j=1}^m |g_{ilj}| \le 1$ for any i and l over $L_{m2}(\boldsymbol{\lambda})$ (or $L_{mh}(\boldsymbol{\lambda})$). Then, we can immediately obtain the surrogate function

$$Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = \sum_{i=1}^n \sum_{l\in\mathcal{Y}} \exp\left[\sum_{j=1}^m \lambda_j(t) g_{ijl}\right] \times \left\{1 - \sum_{j=1}^m |g_{ijl}| + \sum_{j=1}^m |g_{ijl}| \exp\left[\frac{g_{ijl}}{|g_{ijl}|}(\lambda_j - \lambda_j(t))\right]\right\}$$

and the corresponding iterative equation

$$\lambda_s(t+1) = \lambda_s(t) + \frac{1}{2}\ln\left(\frac{\sum_{(i,l)\in S_s^-} |g_{ils}| \exp(\sum_{j=1}^m \lambda_j(t) g_{ijl})}{\sum_{(i,l)\in S_s^+} |g_{ils}| \exp(\sum_{j=1}^m \lambda_j(t) g_{ijl})}\right).$$

We can see that these iterative procedures for binary and multi-class cases are equivalent to those of the parallel-update optimization algorithm of (Collins et al. 2002). However, while ours is built upon the SM algorithm and relies only on the convexity of the exponential function, the one in (Collins et al. 2002) requires the construction of a Bregman distance which is much more mathematically involved. Moreover, convergence of our algorithm follows directly from the SM algorithm because it is obvious that $L_a(\boldsymbol{\lambda})$ or $L_{m2}(\boldsymbol{\lambda})$ ($L_{mh}(\boldsymbol{\lambda})$) is convex in $\boldsymbol{\lambda}$, and $Q_a(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ or $Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$ is continuous in both $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}(t)$. It is worth noting that the Bregman distance optimization algorithm of (Collins et al. 2002) can also work with the first-order Taylor expansion of a convex function. However, the argument of this convex function is itself also a function.

8 SM algorithm for log-linear model

The generalized iterative scaling (GIS) algorithm (Darroch and Ratcliff 1972) is an important method for the log-linear model. In this section we develop an SM algorithm for the log-linear model which can be shown to be equivalent to GIS. Following the notation in (Darroch and Ratcliff 1972), we let I be a finite index set, $\mathbf{p} = \{p_i; i \in I, p_i \ge 0, \sum_{i\in I} p_i \le 1\}$ and $\boldsymbol{\pi} = \{\pi_i; i \in I, \pi_i > 0, \sum_{i\in I} \pi_i \le 1\}$. Now given $\boldsymbol{\pi}$, we seek to find a probability function of the form

$$p_i = \pi_i \prod_{r=1}^c \lambda_r^{a_{ri}}, \tag{41}$$

which satisfies the constraints

$$\sum_{i\in I} a_{ri} p_i = h_r, \quad r = 1, 2, \ldots, c,$$

where a_{ri} and h_r are given and satisfy

$$a_{ri} \ge 0, \quad \sum_{r=1}^c a_{ri} = 1, \qquad h_r > 0, \quad \sum_{r=1}^c h_r = 1.$$

Darroch and Ratcliff (1972) formulated this problem as a constrained minimization problem as follows

$$\min_{\mathbf{p}} \left\{ KL(\mathbf{p}, \boldsymbol{\pi}) = \sum_{i\in I} p_i \ln \frac{p_i}{\pi_i} \right\}, \qquad \text{s.t.} \sum_{i\in I} a_{ri} p_i = h_r, \quad r = 1, \ldots, c.$$

Further, this problem is equivalent to the following unconstrained minimization problem:

$$L(\mathbf{p}, \eta_0, \boldsymbol{\eta}) = \sum_{i\in I} p_i \ln \frac{p_i}{\pi_i} + \sum_{r=1}^{c} \eta_r (a_{ri} p_i - h_r) + \eta_0 \left(\sum_{i\in I} p_i - 1 \right), \tag{42}$$

where η_0 and $\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_c\}$ are the Lagrange multipliers. As

$$\frac{\partial L}{\partial p_i} = \ln \frac{p_i}{\pi_i} + \sum_{r=1}^{c} \eta_r a_{ri} + \eta_0 = 0,$$
$$\frac{\partial L}{\partial \eta_0} = \sum_{i\in I} p_i - 1 = 0,$$

we obtain

$$p_i = \frac{\pi_i \exp(-\sum_{r=1}^{c} \eta_r a_{ri})}{\sum_j \pi_j \exp(-\sum_{r=1}^{c} \eta_r a_{rj})}. \tag{43}$$

Plugging (43) back into (42), we obtain the dual maximization problem (Boyd and Vandenberghe 2004) as

$$F(\boldsymbol{\eta}) = -\sum_r \eta_r h_r - \ln \sum_{i\in I} \left(\pi_i \exp\left(-\sum_r \eta_r a_{ri} \right) \right). \tag{44}$$

Now we apply the SM algorithm to this dual problem. Noticing that both $-\ln(\cdot)$ and $\exp(\cdot)$ are convex, we have

$$\begin{aligned} F(\boldsymbol{\eta}) &\geq -\sum_r \eta_r h_r - \ln \sum_{i\in I} \pi_i \exp\left(-\sum_r \eta_r(t) a_{ri} \right) \\ &\quad - \frac{\sum_{i\in I} \pi_i \exp(-\sum_r \eta_r a_{ri})}{\sum_{i\in I} \pi_i \exp(-\sum_r \eta_r(t) a_{ri})} + 1 \\ &= -\sum_r \eta_r h_r - \ln \sum_{i\in I} \pi_i \exp\left(-\sum_r \eta_r(t) a_{ri} \right) \\ &\quad - \sum_{i\in I} p_i(t) \exp(\sum_r (\eta_r(t) - \eta_r) a_{ri}) \\ &\geq -\sum_r \eta_r h_r - \ln \sum_{i\in I} \pi_i \exp\left(-\sum_r \eta_r(t) a_{ri} \right) \\ &\quad - \sum_{i\in I} p_i(t) \sum_r a_{ri} \exp(\eta_r(t) - \eta_r), \end{aligned}$$

where

$$p_i(t) = \frac{\pi_i \exp(-\sum_{r=1}^{c} \eta_r(t) a_{ri})}{\sum_j \pi_j \exp(-\sum_{r=1}^{c} \eta_r(t) a_{rj})}. \tag{45}$$

This leads us to the $(t+1)$th estimate of η_r, i.e.,

$$\eta_r(t+1) = \eta_r(t) - \ln \frac{h_r}{\sum_{i\in I} p_i(t) a_{ri}}. \tag{46}$$

For $r = 1, \ldots, c$, let $\eta_r(0)$ be equal and randomly generated. We then alternately implement (45) and (46). Recall that the iterative process of GIS for this problem is defined as (Darroch and Ratcliff 1972, Theorem 1)

$$p_i(0) = \pi_i, \qquad p_i(t+1) = p_i(t) \prod_{r=1}^{c} \left(\frac{h_r}{g_r(t)} \right)^{a_{ri}},$$

where $g_r(t) = \sum_{i\in I} a_{ri} p_i(t)$. In fact, with our initial settings on η, it follows easily from (45) that

$$p_i(0) = \frac{\pi_i}{\sum_{j\in I} \pi_j}.$$

Moreover, plugging (46) into (45), we have

$$p_i(t+1) = \frac{p_i(t) \prod_{r=1}^{c} (\frac{h_r}{g_r(t)})^{a_{ri}}}{\sum_{j\in I} p_j(t) \prod_{r=1}^{c} (\frac{h_r}{g_r(t)})^{a_{rj}}},$$

where $g_r(t) = \sum_{i\in I} a_{ri} p_i(t)$. Clearly, our SM algorithm is similar to GIS. However, our SM algorithm satisfies $\sum_{i\in I} p_i(t) = 1$ while GIS only satisfies $\sum_{i\in I} p_i(t) \le 1$. Thus, we may regard our SM algorithm as a variant of GIS that makes the constraint $\sum_{i\in I} p_i = 1$ hold.

9 Concluding remarks

In this paper we have demonstrated the successful application of SM algorithms to generalized linear models, and to the binary logistic regression model in particular. Like EM algorithms for missing data problems, SM algorithms are gaining popularity in computational statistics for problems without missing data. Although EM algorithms have already been commonly used in machine learning, this is currently not the case for SM algorithms. We hope that this paper has successfully demonstrated the power and potential of SM algorithms and will thus lead to its wider adoption in machine learning.

Besides using Jensen's inequality, first-order Taylor approximation or the low quadratic bound principle, we have also demonstrated the possibility of using different combinations of these methods for constructing a surrogate function. In order to deal with multivariable optimization problems, we have also presented CSM and SCM. Furthermore, for this problem we can devise an SCMS algorithm, an alternative based on the idea behind the ECME algorithm (Liu and Rubin 1994), which is an extension of the ECM algorithm (Meng and Rubin 1993). It would be possible to speed up SM algorithms via over-relaxation approaches (Salakhutdinov and Roweis 2003).

Recall that on the one hand, Della Pietra et al. (2001) associated iterative scaling algorithms with an auxiliary function, so iterative scaling algorithms are essentially equivalent to SM algorithms. On the other hand, the Bregman distance-based optimization algorithms (Della Pietra et al. 1997; Kivinen and Warmuth 1999; Lafferty 1999; Collins et al. 2002; Della Pietra et al. 2001) work with the first-order Taylor expansion of a convex function, the argument of which is itself also a function. Therefore, these algorithms also share some common properties with SM algorithms.

Since convexity plays a central role in the methods proposed in this paper, it appears that convexity is a necessary condition for SM algorithms to be applicable. It is worth noting that a recent work (Edwards and Lauritzen 2001) in computational statistics devised a so-called TM algorithm, which alternates between a T-step for calculating a titled version of the unconditional likelihood function and an M-step for maximizing the titled version. The basic idea behind the TM algorithm is to approximate the conditional log-likelihood function by linearizing the corresponding marginal log-likelihood with the first-order Taylor expansion. However, since the TM algorithm does not make use of the convexity property, its convergence is thus not guaranteed. Nevertheless, this algorithm inspires a *convex termination* approach to the applications of SM algorithms in case of non-convexity. For a method designed to work well for a convex function, convex termination refers to the application of this method also to a non-convex function. From this perspective, the TM algorithm has the property of convex termination. This resembles the Newton-like methods that possess the *quadratic termination* property (Fletcher 1987). Thus the work of (Edwards and Lauritzen 2001) sheds some light on the possibility of using SM algorithms for non-convex functions as well. More studies along this line will be pursued in our future work.

Acknowledgements The research reported in this paper has been supported by Competitive Earmarked Research Grants (CERG) 621706 (PI: Dit-Yan Yeung) and 614907 (PI: James T. Kwok) from the Research Grants Council of the Hong Kong Special Administrative Region, China.

Appendix 1: Concavity of the function $f(u) = \ln\cosh(\sqrt{u})$

Lemma 1 *The function*

$$h(x) \equiv \begin{cases} \frac{\tanh(x)}{x}, & x \neq 0, \\ 1, & x = 0. \end{cases}$$

is continuous on $(-\infty, +\infty)$.

Proof If $x \neq 0$, we have

$$\frac{\tanh(x)}{x} = \frac{\exp(x) - \exp(-x)}{x(\exp(x) + \exp(-x))}.$$

Now consider that

$$\lim_{x\to 0} \frac{\exp(x) - \exp(-x)}{x(\exp(x) + \exp(-x))} = \lim_{x\to 0} \frac{\exp(x) + \exp(-x)}{\exp(x) + \exp(-x) + x(\exp(x) - \exp(-x))} = 1,$$

thus $h(x)$ is continuous. □

For $u > 0$,

$$\frac{d\ln\cosh(\sqrt{u})}{du} = \frac{\exp(\sqrt{u}) - \exp(-\sqrt{u})}{2\sqrt{u}(\exp(\sqrt{u}) + \exp(-\sqrt{u}))} = \frac{\tanh(\sqrt{u})}{2\sqrt{u}}.$$

From Lemma 1, we can thus define a continuous function $\varphi(u)$ on $[0,\infty)$ as

$$\varphi(u)\equiv\begin{cases}\frac{\tanh(\sqrt{u})}{2\sqrt{u}}, & u>0,\\ \frac{1}{2}, & u=0.\end{cases}$$

Now we compute the derivative of $\varphi(u)$ on $(0,\infty)$ as

$$\frac{1}{4u}\frac{4\sqrt{u}+\exp(-2\sqrt{u})-\exp(2\sqrt{u})}{(\exp(\sqrt{u})+\exp(-\sqrt{u}))^2}.$$

Since

$$\begin{aligned}\frac{d(2v+\exp(-v)-\exp(v))}{dv}&=2-\exp(-v)-\exp(v)\\&=-(\exp(-v/2)-\exp(v/2))^2<0,\end{aligned}$$

$2v+\exp(-v)-\exp(v)$ is a decreasing function. Hence, we have $4\sqrt{u}+\exp(-2\sqrt{u})-\exp(2\sqrt{u})\le 0$ for $u>0$. Thus, $\varphi(u)$ is decreasing on $(0,\infty)$. Again, using the property that $\varphi(u)$ is continuous on $[0,\infty)$, we have that $\varphi(u)$ is decreasing on $[0,\infty)$. Furthermore, $\varphi(u)\le\frac{1}{2}$ $(\forall u\in[0,\infty))$. Since, up to an additive constant, we can express

$$f(u)=\int_0^u\varphi(v)dv,\quad u\in[0,\infty),$$

according to Theorem 24.2 in (Rockafellar 1970), we obtain that $f(u)$ is a well-defined closed proper concave function on $[0,\infty)$. Moreover, $f'_+(0)=\frac{1}{2}$.

Appendix 2: Proof of the relationship (30)

Lemma 2 *Suppose that $\eta_j\ge 0$ for $j=1,\dots,r$ and $\sum_{j=1}^r\eta_j\le 1$. Let $\boldsymbol{\eta}=(\eta_1,\dots,\eta_r)^T$. Then*

$$\operatorname{diag}(\boldsymbol{\eta})-\boldsymbol{\eta}\boldsymbol{\eta}^T\succeq 0.$$

Proof For an arbitrary $\mathbf{x}=(x_1,\dots,x_r)^T\neq 0\in\mathbb{R}^r$, we have

$$\mathbf{x}^T(\operatorname{diag}(\eta_1,\dots,\eta_r)-\boldsymbol{\eta}\boldsymbol{\eta}^T)\mathbf{x}=\sum_{j=1}^r\eta_jx_j^2-\left(\sum_{j=1}^r\eta_jx_j\right)^2\ge 0.$$

Here we use that the function u^2 is convex on $\mathbb{R}$. □

In this appendix, we want to prove that

$$Q_n(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))\le Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))\le Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))\le Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)).$$

Let $p_i=p_i(\boldsymbol{\lambda}(t))$, $u_i=\boldsymbol{\lambda}^T\mathbf{g}_i$ and $v_i=\boldsymbol{\lambda}(t)^T\mathbf{g}_i$. Then

$$Q_n(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))=L(\boldsymbol{\lambda}(t))+\sum_{i=1}^n(u_i-v_i)p_i+\frac{1}{2}\sum_{i=1}^n p_i(1-p_i)(u_i-v_i)^2,$$

$$Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = L(\boldsymbol{\lambda}(t)) + \sum_{i=1}^{n} \frac{u_i - v_i}{2} + \sum_{i=1}^{n} \frac{\beta_i(t)}{4}(u_i - v_i)(u_i + v_i),$$

$$Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = L(\boldsymbol{\lambda}(t)) + \sum_{i=1}^{n}(u_i - v_i)p_i + \frac{1}{8}\sum_{i=1}^{n}(u_i - v_i)^2.$$

If $v_i = 0$, then $\beta_i = \frac{1}{2}$ and $p_i = \frac{1}{2}$ and so the relationship (30) holds. Now, consider the case where $v_i \neq 0$. Since

$$\begin{aligned}
\beta_i(t) &= \frac{\tanh(|\boldsymbol{\lambda}(t)^T \mathbf{g}_i|/2)}{|\boldsymbol{\lambda}(t)^T \mathbf{g}_i|} \\
&= \frac{\exp(|\boldsymbol{\lambda}(t)^T \mathbf{g}_i|) - 1}{|\boldsymbol{\lambda}(t)^T \mathbf{g}_i|(\exp(|\boldsymbol{\lambda}(t)^T \mathbf{g}_i|) + 1)} \\
&= \frac{\exp(|v_i|) - 1}{|v_i|(\exp(|v_i|) + 1)} \\
&= \frac{\exp(v_i) - 1}{v_i(\exp(v_i) + 1)} \left(\frac{\exp(x) - 1}{x(\exp(x) + 1)} \text{ is even on } (-\infty, 0) \cup (0, +\infty) \right),
\end{aligned}$$

we have

$$\begin{aligned}
&\frac{u_i - v_i}{2} + \frac{\beta_i}{4}(u_i - v_i)(u_i + v_i) - p_i(u_i - v_i) \\
&= \frac{u_i - v_i}{2} + \frac{\exp(v_i) - 1}{4v_i(\exp(v_i) + 1)}(u_i - v_i)(u_i + v_i) - \frac{\exp(v_i)}{1 + \exp(v_i)}(u_i - v_i) \\
&= \frac{2v_i \exp(v_i) + 2v_i + (\exp(v_i) - 1)(u_i + v_i) - 4v_i \exp(v_i)}{4v_i(\exp(v_i) + 1)}(u_i - v_i) \\
&= \frac{\exp(v_i) - 1}{4v_i(\exp(v_i) + 1)}(u_i - v_i)^2.
\end{aligned}$$

From Appendix 1, we know that

$$\frac{\exp(v_i) - 1}{4v_i(\exp(v_i) + 1)} \leq \frac{1}{8}.$$

This then follows that

$$Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) - Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = \sum_{i=1}^{n} \left(\frac{\exp(v_i) - 1}{4v_i(\exp(v_i) + 1)} - \frac{1}{8} \right)(u_i - v_i)^2 \leq 0.$$

On the other hand, consider

$$\begin{aligned}
\phi(v_i) &= \frac{\exp(v_i) - 1}{4v_i(\exp(v_i) + 1)} - \frac{1}{2} p_i(1 - p_i) \\
&= \frac{\exp(v_i) - 1}{4v_i(\exp(v_i) + 1)} - \frac{1}{2} \frac{\exp(v_i)}{1 + \exp(v_i)} \frac{1}{1 + \exp(v_i)} \\
&= \frac{\exp(2v_i) - 1 - 2v_i \exp(v_i)}{4v_i(\exp(v_i) + 1)}.
\end{aligned}$$

For $v_i > 0$, since

$$\frac{d(\exp(2v_i) - 1 - 2v_i \exp(v_i))}{dv_i} = 2\exp(v_i)(\exp(v_i) - 1 - v_i) \geq 0,$$

then $\exp(2v_i) - 1 - 2v_i \exp(v_i) \geq 0$. So $\phi(v_i) \geq 0$. Clearly, $\phi(v_i)$ is even on $(-\infty, 0) \cup (0, +\infty)$. This shows that $\phi(v_i) \geq 0$ on $(-\infty, 0) \cup (0, +\infty)$. Thus, immediately we obtain

$$Q_f(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) - Q_n(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) = \sum_{i=1}^{n}\left(\frac{\exp(v_i) - 1}{4v_i(\exp(v_i) + 1)} - \frac{1}{2}p_i(1 - p_i)\right)(u_i - v_i)^2 \geq 0.$$

We now prove that $Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \leq Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t))$. By Lemma 2, we first have $\operatorname{diag}(|g_{i1}|, \ldots, |g_{im}|) - \mathbf{g}_i\mathbf{g}_i^T$ is positive semidefinite because $\sum_{j=1}^{m} |g_{ij}| \leq 1$. It then follows that

$$\begin{aligned} &Q_m(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) - Q_q(\boldsymbol{\lambda}|\boldsymbol{\lambda}(t)) \\ &\quad = \frac{1}{8}(\boldsymbol{\lambda} - \boldsymbol{\lambda}(t))^T \sum_{i=1}^{n}(\operatorname{diag}(|g_{i1}|, \ldots, |g_{im}|) - \mathbf{g}_i\mathbf{g}_i^T)(\boldsymbol{\lambda} - \boldsymbol{\lambda}(t)) \geq 0. \end{aligned}$$

References

Becker, M. P., Yang, I., & Lange, K. (1997). EM algorithms without missing data. *Statistical Methods in Medical Research*, *6*, 38–54.

Böhning, D., & Lindsay, B. G. (1988). Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, *40*(4), 641–663.

Borg, I., & Groenen, P. (1997). *Modern multidimensional scaling*. New York: Springer.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Collins, M., Schapire, R. E., & Singer, Y. (2002). Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, *47*(2–3), 253–285.

Craven, M., Dopasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the World Web Wide. In *The fifteenth national conference on artificial intelligence*.

Darroch, J. N., & Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, *43*(5), 1470–1480.

De Leeuw, J. (1994). Block relaxation algorithms in statistics. In H. H. Bock, W. Lenski, & M. M. Richter (Eds.), *Information systems and data analysis* (pp. 308–325). Berlin: Springer.

Della Pietra, S., Della Pietra, V., & Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(4), 380–393.

Della Pietra, S., Della Pietra, V., & Lafferty, J. (2001). *Duality and auxiliary functions for Bregman distances* (Technical Report CMU-CS-01-109), School of Computer Science, CMU.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, *39*(1), 1–38.

De Pierro, A. R. (1995). A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Transactions on Medical Imaging*, *14*(1), 132–137.

Edwards, D., & Lauritzen, S. L. (2001). The TM for maximising a conditional likelihood function. *Biometrika*, *88*(4), 961–972.

Fletcher, R. (1987). *Practical methods of optimization* (2nd ed.). New York: Wiley.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, *28*(2), 337–374.

Jaakkola, T., & Jordan, M. (1997). A variational approach to Bayesian logistic regression models and their extensions. In *The sixth international workshop on artificial intelligence and statistics*.

Jebara, T., & Pentland, A. (1999). Maximum conditional likelihood via bound maximization and the CEM algorithm. In *Advances in neural information processing systems* (Vol. 11).

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *The fourteenth international conference on machine learning* (pp. 143–151). San Francisco: Kaufmann.

Kivinen, J., & Warmuth, M. K. (1997). Boosting as entropy projection. In *The twelfth annual conference on computational learning theory* (pp. 134–144).

Lafferty, J. (1999). Additive models, boosting and inference for generalized divergences. In *The twelfth annual conference on computational learning theory* (pp. 125–133).

Lange, K. (1995). A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society Series B*, *57*(2), 425–437.

Lange, K., Hunter, D. R., & Yang, I. (2000). Optimization transfer using surrogate objective functions with discussion. *Journal of Computational and Graphical Statistics*, *9*(1), 1–59.

Lebanon, G., & Lafferty, J. (2001). *Boosting and maximum likelihood for exponential models* (Technical Report CMU-CS-01-144), School of Computer Science, Carnegie Mellon University.

Liu, C., & Rubin, D. B. (1994). The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Bionmetrika*, *84*(4), 633–648.

Meng, X.-L. (2000). Discussion on “optimization transfer using surrogate objective functions”. *Journal of Computational and Graphical Statistics*, *9*(1), 35–43.

Meng, X.-L., & Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Bionmetrika*, *80*(2), 267–278.

Minka, T. P. (2003). *A comparison of numerical optimizers for logistic regression* (Technical report). Available from http://www.stat.cmu.edu/~minka/papers/logreg/.

Ostrowski, A. M. (1960). *Solution of equations and systems of equations*. New York: Academic Press.

Rockafellar, T. (1970). *Convex analysis*. Princeton: Princeton University Press.

Salakhutdinov, R., & Roweis, S. (2003). Adaptive overrelazed bound optimization methods. In *The 20th international conference on machine learning*.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*, 297–336.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics*, *11*, 95–103.

Yuille, A., & Rangarajan, A. (2001). The convex-concave computational procedure (CCCP). In *Advances in neural information processing systems* (Vol. 13).