J. R. Statist. Soc. B (2007)
69, *Part* 2, *pp.* 185–198

On fast computation of the non-parametric maximum likelihood estimate of a mixing distribution

Yong Wang

University of Auckland, New Zealand

[Received January 2006. Revised October 2006]

Summary. A fast algorithm for computing the non-parametric maximum likelihood estimate of a mixing distribution is presented. At each iteration, the algorithm adds new important points to the support set as guided by the gradient function, updates all mixing proportions via a quadratically convergent method and discards redundant support points straightaway. With its convergence being theoretically established, numerical studies show that it is very fast and stable, compared with several other algorithms that are available in the literature.

Keywords: Constrained optimization; Mixture models; Non-parametric maximum likelihood computation; Quadratic approximation; Vertex direction method; Vertex exchange method

1. Introduction

In this paper, we study the computation of the non-parametric maximum likelihood estimate (NPMLE) of a mixing distribution. The density of such a mixture model is of the form

$$f(x; G) = \int_{\Omega} f(x; \theta)\, \mathrm{d}G(\theta), \tag{1}$$

where $f(x; \theta)$, $x \in \mathcal{X}$, $\theta \in \Omega \subset \mathbb{R}$, is the component density and $G(\theta)$ the mixing distribution function. There are a wide range of practical applications for this type of model, in, for example, population heterogeneity studies, non-parametric empirical Bayes estimation and semiparametric density estimation; see Lindsay (1995), Lindsay and Lesperance (1995), Böhning (2000) and the references therein.

Given a random sample $x_1, \ldots, x_n$ from density (1), the log-likelihood of G has the form

$$l(G) = \sum_{i=1}^{n} \log\left\{ \int_{\Omega} f(x_i; \theta)\, \mathrm{d}G(\theta) \right\}. \tag{2}$$

The NPMLE $\hat{G}$ maximizes $l(G)$ among all distribution functions that are defined on Ω and is known to be discrete with support set containing no more points than the number of distinct values in the sample (Laird, 1978; Lindsay, 1983). For a discrete G, let us write $G(\theta) = \Sigma_{j=1}^{m} \pi_j \delta_{\theta_j}$, where $\theta_j \in \Omega$ and $\pi_j > 0$ for $j = 1, \ldots, m$, $\Sigma_{j=1}^{m} \pi_j = 1$, and δ_{θ_j} puts mass 1 at θ_j. Denoting $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)^{\mathrm{T}}$ and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)^{\mathrm{T}}$, we may rewrite density (1) as

$$f(x; \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{j=1}^{m} \pi_j\, f(x; \theta_j), \tag{3}$$

Address for correspondence: Yong Wang, Department of Statistics, University of Auckland, Private Bag 92019, Auckland, New Zealand.
E-mail: yongwang@stat.auckland.ac.nz

 1369–7412/07/69185

and log-likelihood (2) as

$$l(\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log\Big\{ \sum_{j=1}^{m} \pi_j \, f(x_i; \theta_j) \Big\}. \tag{4}$$

Finding $\hat{G}$ is equivalent to finding its $\hat{\boldsymbol{\pi}}$ and $\hat{\boldsymbol{\theta}}$, including their common dimension $\hat{m}$.

One difficulty in computing $\hat{G}$ lies in the fact that $\hat{m}$ is unknown beforehand, so it is not possible to apply the usual optimization techniques directly. There are several methods in the literature for this special optimization problem, including: the expectation–maximization (EM) algorithm (Laird, 1978), the vertex direction method (VDM) (Fedorov, 1972; Wynn, 1970; Wu, 1978a, b), the vertex exchange method (VEM) (Böhning, 1985), the intra-simplex direction method (ISDM) (Lesperance and Kalbfleisch, 1992), the semi-infinite programming (SIP) method (Coope and Watson, 1985; Lesperance and Kalbfleisch, 1992; Susko *et al.*, 1999), Simar's method (Simar, 1976; Böhning, 1982) and the quadratic method (Atwood, 1976). Some of these methods were originally proposed for finding optimal designs of experiments (Silvey, 1980; Pukelsheim, 1993) but are equally applicable to computing NPMLEs. Apart from the EM algorithm, which uses a large number of initial support points, the other methods add one or more new support points that are deemed useful at each iteration of the optimization process. Since the number of support points may increase explosively, to be computationally feasible any such method should use techniques such as keeping no track of support points, collapsing similar ones or discarding bad ones. For an iterative computational method, a critically important issue is the speed of convergence. Many of the aforementioned methods, however, may converge too slowly to be useful in some applications, especially when an NPMLE needs to be computed repeatedly.

Of particular interest in this paper is the quadratic method that was originated by Atwood (1976). This method was given in the context of optimal design of experiments, but like others it can be used for NPMLE computation. Our implementation of this method with some modifications suggests that it converges at a competitively fast, if not faster, speed, as compared with the other existing methods. It appears, however, that this method has been underused for NPMLE computation. In fact, I have not found any numerical studies of this method in the literature of NPMLE computation. Atwood (1976) and Böhning (1985) contained only relatively simple optimal design examples, and in both cases the quadratic method is reported to converge rapidly. This underusage is perhaps partly because Atwood did not specify in detail how the quadratic programming subproblem should be solved, which is critical for a successful and numerically stable implementation, as discussed below.

This paper presents and studies a new algorithm for NPMLE computation, which can be considered as an extension from Atwood's method. The new algorithm solves the quadratic programming subproblem conveniently via a linear regression formulation. Moreover, it adds *many* useful new support points in each iteration, instead of just one, as in Atwood's method, and discards unwanted ones quickly, whereas Atwood collapses similar ones. Specific details of our implementations of both the new algorithm and a modified version of Atwood's method are given. Numerical studies show that the new algorithm usually converges at least several times faster than Atwood's, and many times faster than other methods in the literature. The convergence of the new algorithm is also established, in a proof that depends on much weaker conditions than are required by Atwood's algorithm. A consequence of the proof is that collapsing similar support points is unnecessary for the new algorithm. Nevertheless, since the new algorithm can quickly detect and discard redundant support points, the support set is usually small.

Throughout the paper, we shall frequently use a bold, lower-case letter, say **a**, for a column vector with elements $a_1, \ldots, a_l$, where the length l is often unspecified but should be clear from the context.

The remainder of the paper is organized as follows. As a critical ingredient of NPMLE computation, computing mixing proportions will be first discussed in Section 2. The new algorithm for NPMLE computation is presented in Section 3. Section 4 contains the theoretical work establishing the convergence of the algorithm. Numerical studies are given in Section 5, which compare the performance of several algorithms for computing NPMLEs.

The data and program that was used to analyse them can be obtained from

`http://www.blackwellpublishing.com/rss`

2. Computing mixing proportions

The computation of the vector of mixing proportions, $\boldsymbol{\pi}$ in density (3), is critical for finding an NPMLE. This section studies how to compute $\boldsymbol{\pi}$ iteratively at the quadratic order of convergence when $\boldsymbol{\theta}$ is fixed. The algorithm that is studied here for computing $\boldsymbol{\pi}$ applies to the more general situation when all component densities are completely specified, including when they do not belong to the same distribution family. Estimation of mixing proportions is useful in its own right. McLachlan and Basford (1988) have a chapter devoted to it; see also a relatively recent work by Pilla and Lindsay (2001), who described a strategy of pairing neighbouring densities and rotating the pairing to speed up the convergence of the conventional EM algorithm for computing mixing proportions.

With known $\boldsymbol{\theta}$, $\boldsymbol{\pi}$ is the only unknown. We shall use ∇ for the vector of first-derivative operators and ∇^2 for the matrix of second-derivative operators, with respect to $\boldsymbol{\pi}$ *only*. Let $\boldsymbol{\pi}'$ be an updating vector of $\boldsymbol{\pi}$ and $\boldsymbol{\eta} = \boldsymbol{\pi}' - \boldsymbol{\pi}$. Denote

$$\begin{aligned} \mathbf{s}_i \equiv \mathbf{s}_i(\boldsymbol{\pi}, \boldsymbol{\theta}) &\equiv \nabla[\log\{f(x_i; \boldsymbol{\pi}, \boldsymbol{\theta})\}] \\ &= \left(\frac{f(x_i; \theta_1)}{f(x_i; \boldsymbol{\pi}, \boldsymbol{\theta})}, \ldots, \frac{f(x_i; \theta_m)}{f(x_i; \boldsymbol{\pi}, \boldsymbol{\theta})} \right)^{\mathrm{T}}, \end{aligned} \tag{5}$$

and $\mathbf{S}^{\mathrm{T}} \equiv \mathbf{S}(\boldsymbol{\pi}, \boldsymbol{\theta})^{\mathrm{T}} \equiv (\mathbf{s}_1(\boldsymbol{\pi}, \boldsymbol{\theta}), \ldots, \mathbf{s}_n(\boldsymbol{\pi}, \boldsymbol{\theta}))$. Note that

$$\nabla l = \mathbf{S}^{\mathrm{T}} \mathbf{1}, \tag{6}$$

$$\nabla^2 l = -\mathbf{S}^{\mathrm{T}} \mathbf{S}, \tag{7}$$

where $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}}$. Expanding $l(\boldsymbol{\pi}', \boldsymbol{\theta})$ in the Taylor series about $\boldsymbol{\pi}$ to second order and substituting ∇l and $\nabla^2 l$ by using equations (6) and (7) yields the following quadratic approximation to $l(\boldsymbol{\pi}, \boldsymbol{\theta}) - l(\boldsymbol{\pi}', \boldsymbol{\theta})$:

$$Q(\boldsymbol{\pi}' | \boldsymbol{\pi}, \boldsymbol{\theta}) \equiv -\mathbf{1}^{\mathrm{T}} \mathbf{S} \boldsymbol{\eta} + \frac{1}{2} \boldsymbol{\eta}^{\mathrm{T}} \mathbf{S}^{\mathrm{T}} \mathbf{S} \boldsymbol{\eta} \tag{8}$$

$$= \frac{1}{2} \| \mathbf{S} \boldsymbol{\eta} - \mathbf{1} \|^2 - \frac{n}{2} \tag{9}$$

$$= \frac{1}{2} \| \mathbf{S} \boldsymbol{\pi}' - \mathbf{2} \|^2 - \frac{n}{2}, \tag{10}$$

where $\mathbf{2} = (2, \ldots, 2)^{\mathrm{T}}$ and $\| \cdot \|$ denotes the L_2-norm. Therefore, in the neighbourhood of $\boldsymbol{\pi}$, maximizing $l(\boldsymbol{\pi}', \boldsymbol{\theta})$ with respect to $\boldsymbol{\pi}'$ can be approximately replaced with the following linear regression problem with equality and inequality constraints:

$$\min_{\pi'} \|\mathbf{S}\pi' - \mathbf{2}\|^2, \qquad \text{subject to } \pi'^{\mathrm{T}}\mathbf{1} = 1, \pi' \geqslant \mathbf{0}, \tag{11}$$

where $\mathbf{0} = (0, \ldots, 0)^{\mathrm{T}}$ and the comparison 'greater than or equal to' is elementwise. Note that if π is iteratively updated by minimizing $Q(\pi'|\pi, \theta)$ with θ fixed, the algorithm has the quadratic order of convergence of Newton's method.

There are many algorithms for solving problem (11). One example is the LSEI algorithm in Lawson and Hanson (1974). This algorithm solves a least squares linear regression problem with equality and inequality constraints by transforming it into a least squares linear regression problem with only non-negativity constraints, with the latter being solved by using an active set algorithm called NNLS (see Appendix A). According to our computing experience, LSEI generally works fine for computing π', but handling small values in its solution does not seem to be easy. These values correspond to either exactly zero or small positive masses but may even turn out to be negative. Finding the exact zeros is a useful feature for discarding excessive support points when computing an NPMLE, and computing small positive masses accurately is also numerically critical for the convergence of NPMLE algorithms developed below.

Our implementation uses a numerically more stable method due to Haskell and Hanson (1981) for solving problem (11). According to their theoretic developments, the solution to the least squares problem with non-negativity constraints

$$\min_{\pi'} |\pi'^{\mathrm{T}}\mathbf{1} - 1|^2 + \gamma\|\mathbf{S}\pi' - \mathbf{2}\|^2, \qquad \text{subject to } \pi' \geqslant \mathbf{0}, \tag{12}$$

converges to the solution to problem (11), as $\gamma \to 0+$. The new problem (12) can then be solved by using the NNLS algorithm. In practice, the value of γ should be carefully chosen: too big a value results in $\pi'^{\mathrm{T}}\mathbf{1} = 1$ being badly satisfied, and too small a value may have $\|\mathbf{S}\pi' - \mathbf{2}\|^2$ not properly minimized. It appears to us that using $\gamma = n \times 10^{-6}$, followed with a normalization of the solution vector π', works quite satisfactorily. Hanson and Haskell (1982) also provided a Fortran implementation of this algorithm. Their default choice for the value of γ appears to be too small for the purposes here and $\|\mathbf{S}\pi' - \mathbf{2}\|^2$ can sometimes be improperly minimized near the NPMLE, leading to difficulty in finding an accurate NPMLE solution.

It is worth mentioning that solving problem (11) through problem (12) rather than the LSEI algorithm is purely for numerical reasons. An algorithm such as LSEI that solves problem (11) by turning the equality constraint into an inequality constraint via variable elimination inevitably produces, owing to subtraction, remarkable rounding errors in small or zero π_j-values. In our experience, values of magnitude less than 10^{-6}, positive or negative, in the solution that is provided by LSEI have hardly any significant digits (in double-precision computation). If such small values are truncated to zero, it may occasionally make the NPMLE algorithms that are described below fail to converge. A failure can occur, even when the final NPMLE solution does not contain support points of small masses but intermediate ones do. By contrast, problem (12) has only non-negativity constraints, and no subtraction between similar values is involved during the computation of the NNLS algorithm. The algorithms for NPMLE computation based on problem (12) are hence numerically more stable; we have not experienced any divergence.

To ensure the monotonic increase of log-likelihood at each iteration, we use the back-tracking line search strategy guarded by the Armijo rule, i.e. denoting by η the constrained solution to minimizing $Q(\pi + \eta|\pi, \theta)$, the inequality

$$l(\pi + \sigma^k\eta, \theta) \geqslant l(\pi, \theta) + \alpha\sigma^k \nabla l(\pi, \theta)^{\mathrm{T}}\eta, \qquad 0 < \alpha < \tfrac{1}{2}, \tag{13}$$

is tested in the order $k = 0, 1, 2, \ldots$ until it is first satisfied. The resulting vector $\pi + \sigma^k\eta$ is then chosen to be the new π in the next iteration. The value of α is often chosen to be small. In particular, we exclude α-values between $\frac{1}{2}$ and 1, since such an α-value leaves out the optimal

solution when the log-likelihood is well approximated quadratically. The popular step halving strategy will be used in the numerical studies below, which sets $\sigma = \frac{1}{2}$.

Apart from the above linear regression formulation, there appears to be another subtle difference here from Atwood's (1976) developments for computing $\boldsymbol{\pi}$. He suggested an optimum line search, whereas the Armijo search is used here. Using the inexact Armijo search not only is computationally cheaper but also offers additional benefits for NPMLE computation. The direct quadratic solution is often nearly optimal and an exact line search may only increase the log-likelihood marginally. More importantly, if the direct quadratic solution has many exactly zero entries, using it reduces the size of the support set and thus the computational costs in the subsequent iteration(s).

We shall simply call the above method for computing $\boldsymbol{\pi}'$ and the associated Armijo search the *constrained Newton* (CN) method.

3. Non-parametric maximum likelihood estimate computation

Many algorithms for computing NPMLEs, including the algorithm that is presented below, need to use the directional derivatives of the log-likelihood. Consider two mixing distribution functions $G(\theta)$ and $H(\theta)$, $\theta \in \Omega$. The directional derivative from G to H is defined as

$$d(H;G) \equiv \left.\frac{\partial l\{(1-\varepsilon)G + \varepsilon H\}}{\partial \varepsilon}\right|_{\varepsilon=0} = \sum_{i=1}^{n} \frac{f(x_i;H)}{f(x_i;G)} - n. \tag{14}$$

If $H = \delta_\theta$, we also write $d(H;G)$ as $d(\theta;G)$, which is known as the gradient function. For an arbitrary H, it holds that

$$d(H;G) = \int d(\theta;G)\,\mathrm{d}H(\theta).$$

The gradient function characterizes the NPMLE $\hat{G}$, owing to the celebrated *general equivalence theorem*:

$$\hat{G} \text{ maximizes } l(G) \Leftrightarrow \hat{G} \text{ minimizes } \sup_\theta\{d(\theta;G)\} \Leftrightarrow \sup_\theta\{d(\theta;\hat{G})\} = 0.$$

Furthermore, it holds that

$$\sup_\theta\{d(\theta;G)\} \geqslant l(\hat{G}) - l(G). \tag{15}$$

For these theoretical results, see, for example, Lindsay (1995), theorems 19 and 23.

For reviews of the major methods for NPMLE computation, including EM, VDM, VEM, ISDM and SIP, we refer the reader to Lesperance and Kalbfleisch (1992), Lindsay (1995) and Böhning (1995, 2000). If Atwood's (1976) quadratic method is applied to NPMLE computation, it can be summarized as follows, with slight modifications explained below.

Algorithm 1 (CN1O). Set $s = 0$. From an initial estimate G_0 with finite support and $l(G_0) > -\infty$, repeat the following steps.

Step 1: compute $\theta_s^* = \arg\max_{\theta\in\Omega}\{d(\theta;G_s)\}$. If $d(\theta_s^*;G_s) = 0$, stop.
Step 2: set $\boldsymbol{\theta}_s^+ = (\boldsymbol{\theta}_s^{\mathrm{T}}, \theta_s^*)^{\mathrm{T}}$ and $\boldsymbol{\pi}_s^+ = (\boldsymbol{\pi}_s^{\mathrm{T}}, 0)^{\mathrm{T}}$. Denote by $\boldsymbol{\pi}_{s+1}^-$ the constrained solution of minimizing $Q(\boldsymbol{\pi}'|\boldsymbol{\pi}_s^+, \boldsymbol{\theta}_s^+)$. Define G_{s+1}^- that consists of $\boldsymbol{\pi}_{s+1}^-$ and $\boldsymbol{\theta}_s^+$.
Step 3: find $\varepsilon_s \in [0, 1]$ to maximize $l\{G_s + \varepsilon(G_{s+1}^- - G_s)\}$ with respect to ε.
Step 4: set $G_{s+1} = G_s + \varepsilon_s(G_{s+1}^- - G_s)$ and $s = s + 1$.

To guarantee convergence theoretically, Atwood (1976), equation (2.6), used only an interior value for ε_s in step 3 by enforcing an additional constraint. In the context of the NPMLE, this constraint is equivalent to, for every support point θ_{sj} of G_s,

$$\|\boldsymbol{\eta}_s\|\pi_{sj} \geqslant -C\eta_{sj} \tag{16}$$

for some constant $C>0$, where π_{sj} is the mass that is allocated to θ_{sj} and η_{sj} its change given in $\boldsymbol{\eta}_s = \boldsymbol{\pi}^-_{s+1} - \boldsymbol{\pi}^+_s$. To avoid explosive increase in the number of support points, nearby support points must be combined in practice, thus usually giving relatively large masses. By choosing sufficiently small C, Atwood reckoned that constraint (16) can perhaps be ignored in reality.

Nevertheless, the theoretical work in Section 4 shows that constraint (16) is not necessary; nor is the collapsing of nearby support points. Without enforcing constraint (16), the algorithm has a computational advantage of discarding redundant support points at each iteration. This is because, when $\varepsilon_s = 1$ is chosen, G^-_{s+1} replaces G_s completely. Any support point of G^-_{s+1} with zero mass is deemed redundant and can then be discarded.

In what follows, we consider adding many good support points at each iteration. We use the abbreviation *CNM* for the resulting algorithm, where CN stands for the constrained Newton method and M for multiple support points being added at each iteration.

Algorithm 2 (CNM). Set $s=0$. From an initial estimate G_0 with finite support and $l(G_0) > -\infty$, repeat the following steps.

Step 1: compute all local maxima $\theta^*_{s1}, \ldots, \theta^*_{sp_s}$ of $d(\theta; G_s)$, $\theta \in \Omega$. If $\max_j\{d(\theta^*_{sj}, G_s)\} = 0$, stop.
Step 2: set $\boldsymbol{\theta}^+_s = (\boldsymbol{\theta}^{\mathrm{T}}_s, \theta^*_{s1}, \ldots, \theta^*_{sp_s})^{\mathrm{T}}$. Compute $\boldsymbol{\pi}^-_{s+1}$ by using one step of the CN method.
Step 3: discard all support points with zero entries in $\boldsymbol{\pi}^-_{s+1}$, which gives $\boldsymbol{\theta}_{s+1}$ and $\boldsymbol{\pi}_{s+1}$ of G_{s+1}. Set $s = s+1$.

The ISDM of Lesperance and Kalbfleisch (1992) also adds many local maxima at each iteration but, to guarantee an ascent direction in likelihood, requires that the gradient values at these points are positive. We do not require this in step 1 of algorithm 2, since including local maxima with negative gradients will not affect the eventual convergence of our algorithm but, instead, often speeds it up.

We use a slight modification of Lesperance and Kalbfleisch's (1992) method for computing local maxima of the gradient function. After evaluating the derivative of the gradient function $d'(\theta; G_s) = \partial d(\theta; G_s)/\partial\theta$ over a fine grid of θ-values, say 100, the univariate Newton method is used for computing the maxima located within the intervals $[a, b]$ that satisfy the conditions $d'(a; G_s) > 0$ and $d'(b; G_s) < 0$. Since in our experience the Newton iterates may fall outside the intervals on rare occasions and fail to converge, bracketing intervals are used. Whenever a Newton iterate falls outside the bracketing interval, the bisection iterate is used instead. The end point of the bracketing interval with the same sign in d' as the iterate is then replaced with the iterate. This combined Newton–bisection method proceeds until an accurate local maximum is obtained.

Since the CNM algorithm can quickly discard redundant support points, the number of support points that are maintained throughout is usually small, typically less than twice the number of support points in the NPMLE.

For reason of comparison, one may want to add to the support set, in step 1 of algorithm 2, only the single point that has the largest gradient value. Let us call the resulting algorithm *CN1*. It then differs from algorithm 1 mainly in whether an Armijo or optimum line search is conducted. This is why we name algorithm 1 *CN1O*, where O stands for optimum. It is expected that the convergence of CN1 and CN1O will be slowed down when the NPMLE has many support points. A numerical comparison of these algorithms and others is given in Section 5.

4. Convergence

The theoretical justification for the convergence of algorithm 2 is given in this section. The proofs of convergence for CN1O and CN1 are similar and are omitted.

The proof of the convergence of algorithm 2 depends on much weaker conditions than those used in Atwood (1976) for optimal design problems. In particular, we do not require $\boldsymbol{\eta}$ to be a stationary point in its direction or require a condition for avoiding small masses in the updated proportion vector, such as condition (2.6) in Atwood (1976). In fact, as mentioned earlier, having zero masses in the updated proportion vector is preferable. A consequence of relaxing the conditions is that collapsing similar support points is unnecessary for algorithm 2. Although collapsing similar support points usually helps to speed up convergence slightly, it runs the risk of failing to produce an accurate solution when the NPMLE contains similar support points or, worse still, failing to converge owing to a potential decrease in likelihood.

As already done in Section 3, we use the following notation when updating $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ of G to $\boldsymbol{\pi}'$ and $\boldsymbol{\theta}'$ of G' (or, similarly, updating $\boldsymbol{\pi}_s$ and $\boldsymbol{\theta}_s$ of G_s to $\boldsymbol{\pi}_{s+1}$ and $\boldsymbol{\theta}_{s+1}$ of G_{s+1}). With some new support points found and included, we denote by $\boldsymbol{\theta}^+$ the expanded support point vector from $\boldsymbol{\theta}$, and by $\boldsymbol{\pi}^+$ the corresponding proportion vector expanded from $\boldsymbol{\pi}$ by including 0s for new support points. For the support point vector $\boldsymbol{\theta}^+$, let $\boldsymbol{\pi}'^-$ denote a new proportion vector. Discarding the support points from $\boldsymbol{\theta}^+$ with zero masses in $\boldsymbol{\pi}'^-$ gives rise to $\boldsymbol{\theta}'$ and $\boldsymbol{\pi}'$. In addition, let $\boldsymbol{\eta} = \boldsymbol{\pi}'^- - \boldsymbol{\pi}^+$, $\mathbf{S}^+ = \mathbf{S}(\boldsymbol{\pi}^+, \boldsymbol{\theta}^+)$ and $\mathcal{G}_0 = \{G : l(G) \geqslant l(G_0) > -\infty\}$.

Assumption 1. $f(x;\theta)$ is bounded above for all $x \in \mathcal{X}$ and $\theta \in \Omega$.

Lemma 1. Under assumption 1, there is an upper bound U such that, for all $G \in \mathcal{G}_0$ and all directions $\boldsymbol{\eta}$,

$$\boldsymbol{\eta}^{\mathrm{T}} \mathbf{S}^{+\mathrm{T}} \mathbf{S}^+ \boldsymbol{\eta} \leqslant U. \tag{17}$$

Proof. For every $i \in \{1, \ldots, n\}$,

$$\begin{aligned} \mathbf{s}_i(\boldsymbol{\pi}^+, \boldsymbol{\theta}^+)^{\mathrm{T}} \boldsymbol{\pi}'^- &= \frac{f(x_i; \boldsymbol{\pi}'^-, \boldsymbol{\theta}^+)}{f(x_i; G)}, \\ &\leqslant \frac{\sup\limits_{x \in \mathcal{X},\, \theta \in \Omega} \{f(x;\theta)\}}{\inf\limits_{G \in \mathcal{G}_0} \{f(x_i; G)\}}, \end{aligned} \tag{18}$$

which is bounded above, since $f(x;\theta)$ is bounded above and, because $l(G_0) > -\infty$, $f(x_i; G)$ is bounded below from 0. Therefore,

$$\begin{aligned} \boldsymbol{\eta}^{\mathrm{T}} \mathbf{S}^{+\mathrm{T}} \mathbf{S}^+ \boldsymbol{\eta} &= \|\mathbf{S}^+ \boldsymbol{\pi}'^- - \mathbf{1}\|^2 \\ &\leqslant \|\mathbf{S}^+ \boldsymbol{\pi}'^-\|^2 + \|\mathbf{1}\|^2 \\ &\leqslant n \max_i [\{\mathbf{s}_i(\boldsymbol{\pi}^+, \boldsymbol{\theta}^+)^{\mathrm{T}} \boldsymbol{\pi}'^-\}^2] + n \end{aligned} \tag{19}$$

is also bounded above, which completes the proof. □

Lemma 2. Under assumption 1, the Armijo search that is described in Section 2 and used in step 2 of algorithm 2 always succeeds within a finite number of steps independent of s.

Proof. Let $\boldsymbol{\eta} = t\mathbf{d}$, where $\mathbf{d}$ is the unit vector in the same direction as $\boldsymbol{\eta}$. Since the constrained minimum point must be located at or before the unconstrained minimum point in this direction, minimizing $\|t\mathbf{S}^+\mathbf{d} - \mathbf{1}\|^2$ with respect to t leads to

$$t \leqslant \mathbf{1}^{\mathrm{T}}\mathbf{S}^{+}\mathbf{d}/\mathbf{d}^{\mathrm{T}}\mathbf{S}^{+\mathrm{T}}\mathbf{S}^{+}\mathbf{d}$$

and, substituting $\mathbf{d}$ by $\boldsymbol{\eta}/t$,

$$\boldsymbol{\eta}^{\mathrm{T}}\mathbf{S}^{+\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta} \leqslant \mathbf{1}^{\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta}. \tag{20}$$

It is readily verifiable that, for sufficiently small $\|\mathbf{S}^{+}\boldsymbol{\eta}\|$, we have the Taylor series expansion

$$l(\boldsymbol{\pi}^{+}+\boldsymbol{\eta},\boldsymbol{\theta}^{+}) - l(\boldsymbol{\pi},\boldsymbol{\theta}) = \mathbf{1}^{\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta} - \tfrac{1}{2}\boldsymbol{\eta}^{\mathrm{T}}\mathbf{S}^{+\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta} + o(\|\mathbf{S}^{+}\boldsymbol{\eta}\|^{2}). \tag{21}$$

Therefore, for any $0<\alpha<\frac{1}{2}$, there is a $\lambda>0$ such that, if $\|\mathbf{S}^{+}\boldsymbol{\eta}\| \leqslant \lambda$, then

$$\begin{aligned} l(\boldsymbol{\pi}^{+}+\boldsymbol{\eta},\boldsymbol{\theta}^{+}) - l(\boldsymbol{\pi},\boldsymbol{\theta}) &\geqslant \mathbf{1}^{\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta} - \tfrac{1}{2}\boldsymbol{\eta}^{\mathrm{T}}\mathbf{S}^{+\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta} - (\tfrac{1}{2}-\alpha)\boldsymbol{\eta}^{\mathrm{T}}\mathbf{S}^{+\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta} \\ &\geqslant \alpha\mathbf{1}^{\mathrm{T}}\mathbf{S}^{+}\boldsymbol{\eta}, \end{aligned} \tag{22}$$

where the last inequality is due to inequality (20). The Armijo rule is satisfied for $k=0$.

When $\|\mathbf{S}^{+}\boldsymbol{\eta}\|>\lambda$, $\|\sigma^{k}\mathbf{S}^{+}\boldsymbol{\eta}\| \leqslant \lambda$ can be satisfied for some $k \geqslant 0$, since $\|\mathbf{S}^{+}\boldsymbol{\eta}\| \leqslant \sqrt{U}$ from lemma 1. We hence need at most

$$\bar{k} \equiv \max\left\{ \left\lceil \log_{\sigma}\left(\frac{\lambda}{\sqrt{U}}\right)\right\rceil, 0\right\}$$

steps for Armijo's rule to be satisfied in all situations. □

Theorem 1. Under assumption 1, suppose that $\{G_s\}$ is any sequence created by algorithm 2 and $\hat{G}$ is the NPMLE. Then $l(G_s)\to l(\hat{G})$ monotonically as $s\to\infty$.

Proof. By construction, $l(G_s)$ increases monotonically and thus must converge to some finite value which is less than or equal to $l(\hat{G})$. From the Armijo search and the non-negative definiteness of $\mathbf{S}_s^{+\mathrm{T}}\mathbf{S}_s^{+}$,

$$\begin{aligned} l(G_{s+1}) - l(G_s) &\geqslant \alpha\sigma^{\bar{k}}\mathbf{1}^{\mathrm{T}}\mathbf{S}_s^{+}\boldsymbol{\eta}_s \\ &\geqslant \alpha\sigma^{\bar{k}}(\mathbf{1}^{\mathrm{T}}\mathbf{S}_s^{+}\boldsymbol{\eta}_s - \tfrac{1}{2}\boldsymbol{\eta}_s^{\mathrm{T}}\mathbf{S}_s^{+\mathrm{T}}\mathbf{S}_s^{+}\boldsymbol{\eta}_s). \end{aligned} \tag{23}$$

Denote by $\boldsymbol{\eta}_{sj}$ the direction from $\boldsymbol{\pi}_s^{+}$ to $\mathbf{e}_j$, a vector whose only non-zero component is the jth component, which is 1. For any $0 \leqslant \varepsilon \leqslant 1$, it holds that

$$\|\mathbf{S}_s^{+}\boldsymbol{\eta}_s - \mathbf{1}\|^{2} \leqslant \|\varepsilon\mathbf{S}_s^{+}\boldsymbol{\eta}_{sj} - \mathbf{1}\|^{2},$$

owing to the optimality of $\boldsymbol{\eta}_s$. Expanding both sides and using inequality (23) and lemma 1 give

$$\begin{aligned} l(G_{s+1}) - l(G_s) &\geqslant \alpha\sigma^{\bar{k}}\varepsilon\left(\mathbf{1}^{\mathrm{T}}\mathbf{S}_s^{+}\boldsymbol{\eta}_{sj} - \frac{\varepsilon}{2}\boldsymbol{\eta}_{sj}^{\mathrm{T}}\mathbf{S}_s^{+\mathrm{T}}\mathbf{S}_s^{+}\boldsymbol{\eta}_{sj}\right) \\ &\geqslant \alpha\sigma^{\bar{k}}\varepsilon\left(\mathbf{1}^{\mathrm{T}}\mathbf{S}_s^{+}\boldsymbol{\eta}_{sj} - \frac{\varepsilon}{2}U\right). \end{aligned} \tag{24}$$

Now let us assume that $\sup_{\theta}\{d(\theta;G_s)\}$ does not approach 0 as $s\to\infty$. There must be infinitely many s such that $\sup_{\theta}\{d(\theta;G_s)\} \geqslant \tau$, for some $\tau>0$. Let the j in inequality (24) correspond to the θ that maximizes $d(\theta;G_s)$, which is in $\boldsymbol{\theta}_s^{+}$ owing to step 1 of the algorithm. Using the fact that $\sup\{d(\theta;G_s)\} = \mathbf{1}^{\mathrm{T}}\mathbf{S}_s^{+}\boldsymbol{\eta}_{sj}$, inequality (24) becomes

$$l(G_{s+1}) - l(G_s) \geqslant \alpha\sigma^{\bar{k}}\varepsilon\left(\tau - \frac{\varepsilon}{2}U\right). \tag{25}$$

Without loss of generality, assume that $\tau/U \leqslant 1$ and choose $\varepsilon = \tau/U$, say. Then

$$l(G_{s+1}) - l(G_s) \geqslant \frac{\alpha\sigma^{\bar{k}}\tau^{2}}{2U}, \tag{26}$$

where the right-hand side is a positive value independent of s. This contradicts the Cauchy property for a convergent sequence. Therefore, $\sup_\theta\{d(\theta; G_s)\} \to 0$ and, from inequality (15), $l(G_s) \to l(\hat{G})$, as $s \to \infty$. This completes the proof of the theorem.

5. Numerical studies

This section provides some numerical evidence for the performance of different NPMLE algorithms, including the EM, VDM, VEM, ISDM, CN1O, CN1 and CNM algorithms. Two practical applications are considered. One is from Böhning (2000), where a Poisson mixture with a non-parametric mixing distribution is fitted to a data set with discrete values. The other is from Efron (2004) and is concerned with fitting a normal mixture with unit component variance to some microarray DNA expression data. In addition, simulated data sets were generated from a model that was suggested by Efron and were used for comparing the speed of convergence of various algorithms. Only the ISDM, CN1O, CN1 and CNM algorithms are sufficiently fast to be included in the simulation study. It can be seen below that CNM is the fastest and most stable algorithm in all cases studied.

A flat likelihood surface in the neighbourhood of the NPMLE may potentially have an adverse effect on an NPMLE algorithm (Böhning (2000), section 2.6, for example). It may cause it to converge extremely slowly and to have difficulty in producing an accurate solution. It should be noted that the 'inaccuracy' here may be only in the sense of G, but not in the sense of $l(G)$. It can be seen from the studies below, and in many other studies that we conducted but have not presented here, that the effect of the flatness of the likelihood surface on the three algorithms CN10, CN1 and CNM is indiscernible. This can be explained by their use of the correct quadratic order information about the likelihood function of $\boldsymbol{\pi}$.

All the algorithms are implemented in R (R Development Core Team, 2004), except that the Fortran implementation of the NNLS algorithm that is needed internally by CN1O, CN1 and CNM was downloaded from `http://www.netlib.org/lawson-hanson`. All computation was done on a laptop computer with a 1.6 GHz Intel Pentium M central processor unit (CPU). The R function `optimize` is used for the optimum line search in the CN1O algorithm, with the setting `tol=0.001`, when $\varepsilon = 1$, corresponding to the quadratic solution, is not optimal already over $[0, 1]$.

5.1. Example 1

We consider the data set that was used in example 1.2 of Böhning (2000). The data, which are reproduced in Table 1, were collected in a cohort study in north-east Thailand in which 602 preschool children participated. For each child, the number of illness spells x, such as fever, cough or running nose, is recorded for all 2-week periods from June 1982 to September 1985. The empirical density (or mass function) is shown as a bar graph in Fig. 1. The MLE for a single Poisson distribution is a poor fit, as commented by Böhning (2000), owing to overdispersion. Using the Poisson mixture with a non-parametric mixing distribution seems more appropriate. For density (1), the component density here is

$$f(x;\theta) = \exp(-\theta)\frac{\theta^x}{x!}, \qquad x \in \{0, 1, 2, \ldots\}, \quad \theta \geqslant 0. \tag{27}$$

The NPMLE $\hat{G}$ which was found by the CNM algorithm is $\hat{\boldsymbol{\pi}} = (0.1969, 0.4800, 0.2693, 0.0538)^{\mathrm{T}}$ and $\hat{\boldsymbol{\theta}} = (0.1434, 2.8173, 8.1642, 16.1558)^{\mathrm{T}}$, after rounding to four decimal places. The computation took 20 iterations and 0.27 s. The maximum gradient at the NPMLE is 2.73×10^{-7}, which is achieved at $\theta = 0.1433$. The gradient curve at this solution is shown in Fig. 2, and the fitted Poisson mixture is plotted in Fig. 1.

Table 1. Thailand data set, with the number of illness spells x and the frequency (i.e. number of children)

	Frequency for the following values of x:									
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
$\geqslant 0$	120	64	69	72	54	35	36	25	25	19
$\geqslant 10$	18	18	13	4	3	6	6	5	1	3
$\geqslant 20$	1	2	0	1	2					

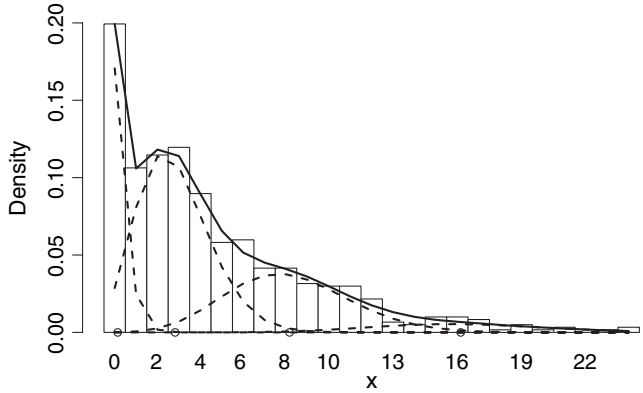

Fig. 1. Empirical density for the Thailand data set (□), and the Poisson mixture using the NPMLE (———): each mixture component weighted with its proportion $\hat{\pi}_j$ is also shown (-------), along with its parameter $\hat{\theta}_j$ (∘).

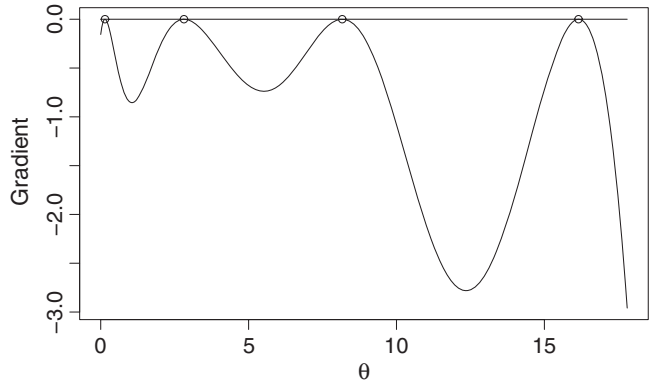

Fig. 2. Gradient curve at the NPMLE for the Thailand data set

The solution that was provided by Böhning (2000), page 7, is $\hat{\boldsymbol{\pi}} = (0.03, 0.17, 0.48, 0.28, 0.05)^{\mathrm{T}}$ and $\hat{\boldsymbol{\theta}} = (0, 0.18, 2.82, 8.20, 16.15)^{\mathrm{T}}$. It is presumably obtained from a combined algorithm of VEM and EM, i.e. use the VEM over a fine grid of values in the first phase to identify support points, and then use the EM algorithm in the second phase for fine-tuning of parameter values; see section 3.4 of Böhning (2000). Despite there being many iterations in each phase, this solution is not as accurate as the solution that is provided by CNM. It has a maximum gradient 30.35, which is achieved at $\theta = 17.83$, and hence does not satisfy very well the NPMLE condition

Table 2. Computation of the NPMLE of the mixing distribution for the Thailand data set

Algorithm	s	$l(\hat{G}) - l(G_s)$	$\sup_\theta\{d(\theta; G_s)\}$	*Time (s)*
VDM	70359	6.76×10^{-3}	1.00×10^{-2}	610.61
VEM	14064	6.14×10^{-3}	9.62×10^{-3}	434.06
EM	5337	1.85×10^{-7}	9.95×10^{-7}	5.58
ISDM	146	2.93×10^{-7}	5.24×10^{-7}	1.46
CN1O	68	5.17×10^{-8}	6.76×10^{-7}	1.35
CN1	56	6.34×10^{-8}	9.90×10^{-7}	0.84
CNM	20	1.18×10^{-9}	2.74×10^{-7}	0.27

that is specified in the general equivalence theorem. Additionally, it contains one more support point than the above solution. The difficulty here for the combined VEM and EM algorithm, and other existing algorithms also, is caused by the flat likelihood surface. Partly to avoid pitfalls like this, Böhning (2000), chapter 4, recommended comparing the NPMLE against estimates with a lower number of components; see Böhning (2003) for a different strategy. In contrast, CNM has no problem in quickly finding an accurate solution, in both the sense of G and of $l(G)$.

The EM, VDM, VEM, ISDM, CN1O, CN1 and CNM algorithms are used for computing the NPMLE. For all of them, the initial support points are chosen to be $\boldsymbol{\theta}_0 = (0, 4, \ldots, 20)^{\mathrm{T}}$ with equal masses assigned. On the basis of inequality (15), the EM, ISDM, CN1O, CN1 and CNM algorithms are stopped when $\sup_\theta\{d(\theta; G_s)\} \leqslant 10^{-6}$ is satisfied, and the VDM and VEM algorithms are stopped at $\sup_\theta\{d(\theta; G_s)\} \leqslant 10^{-2}$ because of their slow convergence. Results are given in Table 2. CNM gives the best performance among all, in terms of both the number of iterations and the execution time.

5.2. Example 2

Efron (2004) developed an empirical Bayes plan for estimating a null hypothesis distribution when a large number of hypotheses are being simultaneously tested. The motivating example that he used was a human immunodeficiency virus drug mutation data set, which was originally from Wu *et al.* (2003). Among the 1391 patients who received at least one of six popular protease inhibitor drugs, 74 positions on the HIV protease gene showed more than three mutations and were thus used in the investigation. Using logistic regression analysis, $444 = 6 \times 74$ z-values were computed, one for testing each null hypothesis, that each drug does not cause mutation at each position. To provide further insights, he fitted an eight-component normal mixture with unit component variance to the histogram counts of the z-values, via a non-linear minimization program and using Poisson deviance as the fitting criterion. The fitted mixing distribution is given in Table 3, where θ_j is the jth component mean. Fig. 3 shows this mixture and a histogram of 1000 values randomly generated from the mixture.

We can also consider using the NPMLE, computed directly from the z-values. Here, instead of using the specific data, we investigate the performance of various algorithms in such a scenario.

Table 3. Estimated mixing distribution in Efron (2004)

$100\pi_j$	1.5	1.3	5.6	12.3	13.6	60.8	2.7	2.2
θ_j	−10.9	−7.0	−4.9	−1.8	−1.1	0.0	2.4	6.1

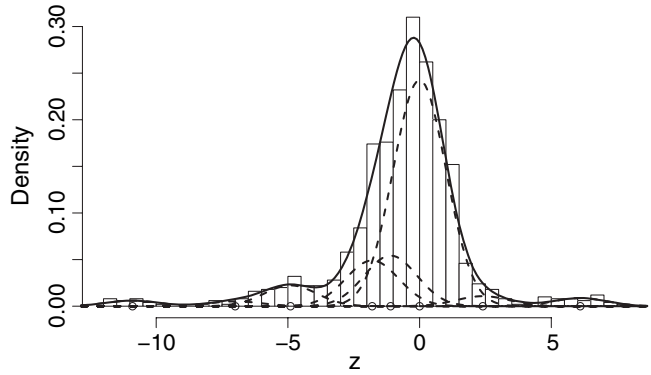

Fig. 3. Mixture density with the parameter values that are given in Table 3 (———), along with its components weighted proportionally (-------), and a histogram of 1000 values that were simulated from the mixture

Table 4. Five-number summaries of the number of iterations and execution times that were required by four algorithms over 100 simulated data sets in example 2

Algorithm	*Numbers of iterations for the following statistics:*					*Times (s) for the following statistics:*				
	Minimum	*Q1*	*Median*	*Q3*	*Maximum*	*Minimum*	*Q1*	*Median*	*Q3*	*Maximum*
ISDM	69	120	235	572	8805	15.1	27.1	51.0	129.1	1923.0
CN1O	40	71	82	95	182	18.6	35.3	43.8	50.4	116.6
CN1	55	70	82	97	172	20.9	28.1	33.6	40.4	81.8
CNM	12	13	14	15	20	4.5	5.5	6.1	6.7	8.5

100 random samples, each of size 1000, are generated from the mixture model with G given in Table 3. Each algorithm is then employed to compute an NPMLE from each sample. Only the ISDM, CN1O, CN1 and CNM algorithms are included in the study; the computational costs are too high to include VDM, VEM or EM. All algorithms are started by taking the mixing distribution that is given in Table 3 as G_0 and terminated when the condition $\sup\{d(\theta; G_s)\} < 10^{-5}$ is satisfied, which gives at least 10-digit accuracy in likelihood.

From the 100 runs of each studied algorithm, a five-number summary is calculated for respectively the number of iterations and execution time, and given in Table 4. It can be seen from the results that CNM is at least several times faster than all the other algorithms. It also appears to be very stable, as it always terminates in at most 20 iterations and its longest execution time is less than twice its shortest time. Although using about the same number of iterations, CN1 is almost always faster than CN1O, which is an indication of the appropriateness of the Armijo search. The ISDM algorithm on average can perhaps be considered as a competitive algorithm with CN1O and CN1, but it does not seem to be very stable. In the worst case, it requires 8805 iterations and/or 1923 s before satisfying the stopping criterion.

Acknowledgements

I thank Alastair Scott and Chris Wild for their constant encouragements and support. Thanks also go to Catherine Loader for advice on revising the first manuscript. The Joint Editor and two referees provided suggestions that led to an improved version. The research is partly supported by a University of Auckland research grant (3605462/9345).

Appendix A: NNLS and related algorithms

The NNLS algorithm of Lawson and Hanson (1974), page 161, is an active set method that solves the least squares linear regression problem with non-negativity constraints (problem NNLS), i.e.

$$\min_{\mathbf{x}}(\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2), \qquad \text{subject to } \mathbf{x} \geqslant \mathbf{0}. \tag{28}$$

Owing to its critical role in our implementation, it is briefly described here. Let $\mathbf{g}(\mathbf{x}) = \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{x}-\mathbf{b})$, which is half the gradient of $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2$.

A.1. Algorithm 3 (NNLS)

Set $\mathbf{x} = \mathbf{0}$, and repeat the following steps.

Step 1: if $g_j(\mathbf{x}) \geqslant 0$ for every j with $x_j = 0$, stop and return $\mathbf{x}$.
Step 2: solve $\min_{\mathbf{z}}(\|\mathbf{A}\mathbf{z}-\mathbf{b}\|^2)$, subject to $z_j = 0$ if $x_j = 0$ and g_j is not the minimum.
Step 3: if $\mathbf{z} \geqslant \mathbf{0}$, set $\mathbf{x} = \mathbf{z}$ and go to step 1.
Step 4: set $\mathbf{x} = \alpha\mathbf{z} + (1-\alpha)\mathbf{x}$, using the maximum $\alpha \in (0, 1)$ that ensures the new $\mathbf{x} \geqslant \mathbf{0}$. Go to step 2.

In this algorithm, step 2 ensures that $\mathbf{z}$ is a least squares solution for non-zero-constrained entries. If $\mathbf{z}$ has no negative entry and thus potentially is a solution to problem (28), it becomes $\mathbf{x}$ in step 3 and the algorithm returns to step 1 for exit testing; otherwise, the non-negative point closest to $\mathbf{z}$ from $\mathbf{x}$ to $\mathbf{z}$ becomes the new $\mathbf{x}$, which because of the optimality of $\mathbf{z}$ must strictly decrease $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2$. In the latter case, if the new $\mathbf{x}$ fails to be a least squares solution as if its zero entries were constrained, another constrained least squares fitting is needed, until such iteratively computed $\mathbf{x}$ is a potential solution to problem (28). The algorithm terminates after step 1, if the gradient vector corresponding to every zero in $\mathbf{x}$ either is zero or points into the non-negative region. Therefore, by satisfying the Karush–Kuhn–Tucker conditions on termination, $\mathbf{x}$ must be a solution to problem (28). Moreover, the algorithm must terminate within a finite number of steps, owing to the finiteness of the number of variables and the fact that $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2$ is strictly decreasing for every new $\mathbf{x}$.

A.2. Related algorithms

Algorithm NNLS is fundamental for other linear regression algorithms that were discussed in Lawson and Hanson (1974). As shown there, a linear regression problem with inequality constraints (problem LSI) can be converted into problem NNLS. If, in addition, the problem has equality constraints (problem LSEI), it can be first converted into problem LSI by eliminating linear dependence between the variables and converting the equality constraints into inequality constraints.

The least squares linear regression problem with non-negativity and equality constraints (problem NNLSE) can be solved by using either the LSEI algorithm or the NNLSE algorithm, i.e. the NNLS algorithm through a formulation that is similar to problem (12). The solution due to NNLSE is numerically more accurate and stable. Owing to the poor conditioning and rounding errors, the LSEI algorithm may map a feasible point in the constrained region to one that is infeasible, or vice versa. See Haskell and Hanson (1981) for detailed analysis of the NNLSE algorithm.

References

Atwood, C. L. (1976) Convergent design sequences, for sufficiently regular optimality criteria. *Ann. Statist.*, **4**, 1124–1138.

Böhning, D. (1982) Convergence of Simar's algorithm for finding the maximum likelihood estimate of a compound Poisson process. *Ann. Statist.*, **10**, 1006–1008.

Böhning, D. (1985) Numerical estimation of a probability measure. *J. Statist. Planng Inf.*, **11**, 57–69.

Böhning, D. (1995) A review of reliable algorithms for the semi-parametric maximum likelihood estimator of a mixture distribution. *J. Statist. Planng Inf.*, **47**, 5–28.

Böhning, D. (2000) *Computer-assisted Analysis of Mixtures and Applications: Meta-analysis, Disease Mapping, and Others*. Boca Raton: Chapman and Hall–CRC.

Böhning, D. (2003) The EM algorithm with gradient function update for discrete mixtures with known (fixed) number of components. *Statist. Comput.*, **13**, 257–265.

Coope, I. D. and Watson, G. A. (1985) A projected Lagrangian algorithm for semi-infinite programming. *Math. Progrmmng*, **32**, 337–356.

Efron, B. (2004) Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *J. Am. Statist. Ass.*, **99**, 96–104.
Fedorov, V. V. (1972) *Theory of Optimal Experiments* (Engl. transl.). New York: Academic Press.
Hanson, R. J. and Haskell, K. H. (1982) Algorithm 587: Two algorithms for the linearly constrained least squares problem. *Ass. Comput. Mach. Trans. Math. Softwr.*, **8**, 323–333.
Haskell, K. H. and Hanson, R. J. (1981) An algorithm for linear least squares problems with equality and nonnegativity constraints. *Math. Progrmmng*, **21**, 98–118.
Laird, N. M. (1978) Nonparametric maximum likelihood estimation of a mixing distribution. *J. Am. Statist. Ass.*, **73**, 805–811.
Lawson, C. L. and Hanson, R. J. (1974) *Solving Least Squares Problems*. Englewood Cliffs: Prentice-Hall.
Lesperance, M. L. and Kalbfleisch, J. D. (1992) An algorithm for computing the nonparametric MLE of a mixing distribution. *J. Am. Statist. Ass.*, **87**, 120–126.
Lindsay, B. G. (1983) The geometry of mixture likelihoods: a general theory. *Ann. Statist.*, **11**, 86–94.
Lindsay, B. G. (1995) *Mixture Models: Theory, Geometry, and Applications*. Hayward: Institute of Mathematical Statistics.
Lindsay, B. G. and Lesperance, M. L. (1995) A review of semiparametric mixture models. *J. Statist. Planng Inf.*, **47**, 29–39.
McLachlan, G. and Basford, K. (1988) *Mixture Models: Inference and Applications to Clustering*. New York: Dekker.
Pilla, R. S. and Lindsay, B. G. (2001) Alternative EM methods for nonparametric finite mixture models. *Biometrika*, **88**, 535–550.
Pukelsheim, F. (1993) *Optimal Design of Experiments*. New York: Wiley.
R Development Core Team (2004) *R: a Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.
Silvey, S. D. (1980) *Optimal Design*. London: Chapman and Hall.
Simar, L. (1976) Maximum likelihood estimation of a compound Poisson process. *Ann. Statist.*, **6**, 1200–1209.
Susko, E., Kalbfleisch, J. D. and Chen, J. (1999) Computational methods for non-parametric maximum likelihood estimation of mixtures. In *Proc. Interface '99*, pp. 432–438. Shaumburg: Interface Foundation of North America.
Wu, C. F. (1978a) Some algorithmic aspects of the theory of optimal designs. *Ann. Statist.*, **6**, 1286–1301.
Wu, C. F. (1978b) Some iterative procedures for generating nonsingular optimal designs. *Communs Statist. Theory Meth.*, **7**, 1399–1412.
Wu, T. D., Schiffer, C. A., Gonzales, M. J., Taylor, J., Kantor, R., Chou, S., Israelski, D., Zolopa, A. R., Fessel, W. J. and Shafer, R. W. (2003) Mutation patterns and structural correlates in human immunodeficiency virus type 1 protease following different protease inhibitor treatments. *J. Virol.*, **77**, 4836–4847.
Wynn, H. P. (1970) The sequential generation of D-optimal experimental design. *Ann. Math. Statist.*, **41**, 1655–1664.