climate of increasingly accessible computing resources and the every increasing numbers of large, complex, and interesting data problems. The section is constantly on the look out for ways to continue to support the efforts of those involved in statistical computing. If anybody has any ideas are suggestions, in particular ideas involving the support and development of students, please let me know.

Editorial Note

This issue of the newsletter features three articles. Jan De Leeuw tells us about generalizations of the EM algorithm. Michael Lawrence and Duncan Temple Lang describe a new R package for writing GUIs, that is portable across platforms. Ivo Dinov reports on his teaching software, Statistical Online Computational Resource, SOCR. Thanks to these authors for their contributions. It would be helpful to have more contributed articles, so please consider contributing your work.

The News section is loaded with information about the upcoming JSM '06 in Seattle. Michael Trosset details the Statistical Computing program and Juergen Symanzik highlights the Graphics program. Don't forget to attend the Joint Computing and Graphics Mixer on Monday starting at 7:30pm. There are always good conversations, food, drinks and lots of door prizes to win. Summaries of two recent conferences, the Interface between Computing Science and Statistics and Fast Manifold Learning, are provided by Yasmin Said and Michael Trosset, respectively.

Finally, there are two informative articles about the field. Tim Hesterberg reports on computing and graphics activities at Insightful. Robert Gould announces the birth of the new free electronic Journal of Technological Innovations in Statistics Education. An analysis of publications in the existing Journal of Statistical Education, conducted with Nathan Yau, shows a need for a publication outlet for technological research.

Di Cook and Juana Sanchez

Featured Article

SOME MAJORIZATION TECHNIQUES

Jan de Leeuw
University of California, Los Angeles
deleeuw@stat.ucla.edu
http://gifi.stat.ucla.edu

1. Introduction

Majorization algorithms (Deleeuw, 1994; Heiser, 1995, Lange et al., 2000) are used with increasing frequency in statistical computation. They generalize the EM algorithm to a much broader class of problems and they can usually be tailored to handle very high-dimensional problems.

The general idea is simple. If we want to minimize f over $X \subseteq R^n$, we construct a *majorization function* g on X × X such that

$$\begin{aligned} f(x) &\leq g(x,y) \qquad \forall x,y \in X \\ f(x) &= g(x,x) \qquad \forall x \in X \end{aligned}$$

Thus g, considered as a function of x is never below f and touches f at y.

The majorization algorithm corresponding with this majorization function g updates x at iteration k by

$$x^{(k+1)} \in \underset{\mathbf{x \in X}}{\mathbf{argmin}}\; g\left(x, x^{(k)}\right),$$

unless we already have

$$x^{(k)} \in \underset{\mathbf{x \in X}}{\mathbf{argmin}}\; g\left(x, x^{(k)}\right),$$

in which case we stop. Convergence follows, under some additional simple conditions, from the *sandwich inequality,* which says that if we do not stop at iteration k, then

$$f\left(x^{(k+1)}\right) \leq g\left(x^{(k+1)}, x^{(k)}\right) < g\left(x^{(k)}, x^{(k)}\right) = f\left(x^{k}\right)$$

Consider the example $f(x)$=1/4 x^4 + 1/2 x^2. The function has local minima at +1 and -1, with function value -1/4, and a local maximum at 0 with function value 0. A

majorization function can be constructed by using $x^2 \geq y^2 + 2y(x-y)$, giving $g(x,y)$=1/4 x^4+ 1/2 y^2 -xy. This leads to the majorization algorithm $x^{(k+1)} = \sqrt[3]{x^{(k)}}$. This converges linearly, with convergence rate 1/3, to either +1 or -1, depending on where we start.

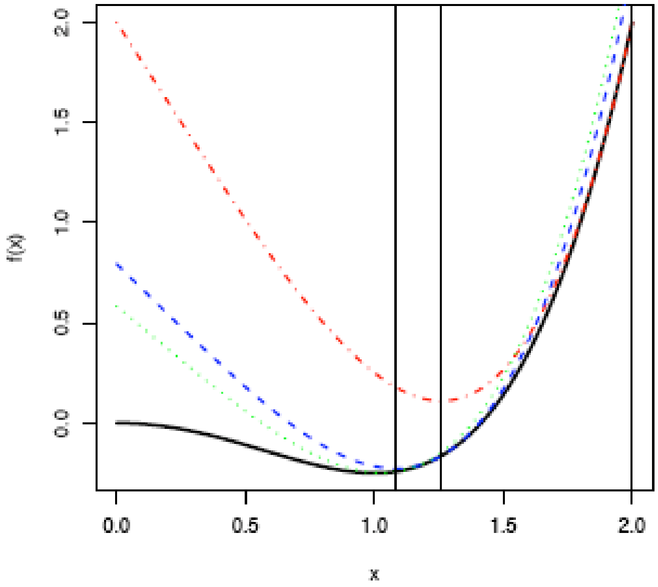

FIGURE 1. Majorization Example

In Figure 1 the function we are minimizing is the thick line.We start at 2. The majorization function at this point, drawn in red with dots-dashes, touches f at 2 and is minimized at $\sqrt[3]{2} \approx 1.2599$. We compute the new majorization function at this point, in blue with dashes, and minimize it at $\sqrt[9]{2} \approx 1.0801$. The next step (green, with dots) takes us to $\sqrt[27]{2} \approx 1.0260$.

Observe that our algorithm is $x^{(k)} = x^{3-k}$. An equally valid majorization algorithm is $x^{(k)} = (-1)^k x^{\frac{1}{3^k}}$, which also minimizes the majorization function in each step. It produces a decreasing sequence of loss function values converging to 1/4, but the sequence of solutions is not convergent and has two converging subsequences, one converging to -1 and one to +1.

In most cases majorization methods converge at a linear rate, with the rate equal to the largest eigenvalue in modulus of the matrix

$$I - \left[D_{11}g(x,x)\right]^{-1} D^2 f(x) = -\left[D_{11}g(x,x)\right]^{-1} D_{12}g(x,x)$$

where the derivatives are evaluated at the fixed point x (Ortega and Rheinboldt, 1970,page 300-301). In some special cases we can have sub-linear or super-linear convergence, but linear convergence is the rule.

2. Using Elementary Inequalities

The first way to construct majorization functions is the simplest one. There are many inequalities in the literature of the form $F(x,y) \geq 0$ with equality if and only if x=y. Such inequalities can often be used to construct majorization functions. Since this is not really a systematic approach, we merely illustrate it by a rather detailed example.

After a suitable choice of coordinates and normalization the Euclidean multidimensional scaling problem can be formulated as minimization of

$$(1) \qquad \sigma(x) = 1 + \frac{1}{2}x'x - \sum_{i=1}^{n} w_i \delta_i d_i(x)$$

Here the w_i are known positive *weights,* the δ_i are the *dissimilarities*, and the $d_i(x)$ are the *Euclidean distances*, defined by $d_i(x) = \sqrt{x' A_i x}$. The A_i are known positive

semi-definite matrices that satisfy $\sum_{i=1}^{n} w_i A_i = I$.

In most cases of interest the dissimilarities will be positive, but we shall cover the more general case in which there can be both positive and negative ones. Decomposing δ_i into its positive and negative parts, i.e. $\delta_i = \delta_i^+ - \delta_i^-$ with both δ_i^+ and δ_i^- non-negative. Now we can write

$$(2) \qquad \sigma(x) = 1 + \frac{1}{2}x'x - \sum_{i=1}^{n} w_i \delta_i^+ d_i(x) + \sum_{i=1}^{n} w_i \delta_i^- d_i(x)$$

If $d_i(y)>o$ then by, respectively, the Cauchy-Schwartz and the Arithmetic-Geometric Mean Inequality

$$\frac{1}{d_i(y)} x' A_i y \le d_i(x) \le \frac{1}{d_i(y)} \frac{1}{2}\left(x' A_i x + y' A_i y\right),$$

Thus

$$(3a) \qquad \sum_{i=1}^{n} w_i \delta_i^+ d_i(x) \ge x' B^+(y) y,$$

with

$$(3b) \qquad \mathrm{B}^+(y) = \sum_{i=1}^{n} w_i \frac{\delta_i^+}{d_i(y)} A_i,$$

And

$$(3c) \qquad \sum_{i=1}^{n} w_i \delta_i^- d_i(x) \le \frac{1}{2}\left(x' B^-(y) x + y' B^-(y) y\right),$$

with

$$(3d) \qquad \mathrm{B}^-(y) = \sum_{i=1}^{n} w_i \frac{\delta_i^-}{d_i(y)} A_i,$$

Observe that both B^+ and B^- are positive semi-definite. Combining these results gives

$$(4) \qquad \sigma(x) \le 1 + \frac{1}{2}x'x - x'B^+(y)y + \frac{1}{2}x'B^-(y)x + \frac{1}{2}y'B^-(y)y.$$

The right-hand side of (4) gives a quadratic majorization function, and the corresponding algorithm is

$$x^{(k+1)} = \left[I + B^-\left(x^{(k)}\right)\right]^{-1} B^+\left(x^{(k)}\right) x^{(k)}.$$

At a stationary point x the derivative of the algorithmic map is

$$\left[I + B^-(x)\right]^{-1}\left[B^+(x) - H^+(x) + H^-(x)\right],$$

where

$$\mathrm{H}^+(x) = \sum_{i=1}^{n} w_i \frac{\delta_i^+}{d_i^3(x)} A_i x x' A_i,$$

and

$$\mathrm{H}^-(x) = \sum_{i=1}^{n} w_i \frac{\delta_i^-}{d_i^3(x)} A_i x x' A_i,$$

The matrices H^+, H^-, and B^+-H^+ are all positive semidefinite.

3. Integrals

Supposed want to maximize

$$f(x) = \log \int_z \exp\{u(x,z)\} dz$$

Because we are maximizing we will now construct a minorization function and a minorization algorithm.

Of course the logarithm in the definition of f is really irrelevant here and the exponent merely guarantees that we are integrating a positive function. Write

$$f(x) - f(y) = \log \frac{\int_Z \exp\{u(y,z)\} \frac{\exp\{u(x,z)\}}{\exp\{u(y,z)\}} dz}{\int_Z \exp\{u(y,z)\} dz}$$

Jensen's inequality, or equivalently the concavity of the logarithm, tells us that

$$f(x) - f(y) \ge \frac{\int_Z \exp\{u(y,z)\} \log \frac{\exp\{u(x,z)\}}{\exp\{u(y,z)\}} dz}{\int_Z \exp\{u(y,z)\} dz}$$

Define

$$\pi(z \mid y) = \frac{\exp\{u(z,y)\}}{\int_Z \exp\{u(z,y)\}}.$$

Then

which defines our minorization function.

A step of the minorization algorithm simply maximizes (in the "M'' step) the ``expectation'' $\int_Z \pi(z \mid y) u(x,z) dz.$

Computing, and possibly simplifying, this expectation is the ``E" step. The algorithm is especially attractive, of course, if the integral defining the expectation can be evaluated in closed form. This is often the case in exponential family problems in statistics, where we want to compute maximum likelihood estimates. It is hardly necessary to give an example in this case, because so many examples of the EM algorithmare available.

4. Using Convexity

Suppose we want to minimize *f(x)* on a convex set *X*. Under very general conditions we can write *f* as the difference of two convex functions. It is sufficient to assume, for example, that *f* is twice continuously differentiable. It is necessary and sufficient that *f* is the indefinite integral of a function of locally bounded variation (Hartman,1959).

If *f=u-v,* with *u* and *v* both convex, then we use

$$v(x) \geq v(y) + Dv(y)(x-y)$$

to construct the convex majorization function

$$g(x,y) = u(x) - v(y) - Dv(y)(x-y).$$

The majorization method reduces optimization of an arbitrary function to solving a sequence of convex optimization problems. Of course matters simplify if *u(x)* can be chosen to be quadratic. For this majorization we find for the derivative of the algorithmic map

$$\left[D^2 u(x)\right]^{-1} D^2 v(x).$$

5. Using Taylor's Theorem

By Taylor's theorem

$$(5)\ f(x) \leq f(y) + (x-y)' Df(y) + \frac{1}{2} \max_{0 \leq \xi \leq 1} (x-y)' D^2 f(\xi x + (1-\xi) y)(x-y),$$

and the right hand side can be used as the majorization function. Of course this general approach can also be applied if we only use the linear term in the Taylor expansion, and also if we use third or higher order terms (De Leeuw, 2006). And by replacing max by min we can use it to construct minorization functions.

But let us continue with *quadratic majorization*. The majorization function in (5) is not necessarily simple, so we may want one that is easier to compute. Suppose there is a matrix *B* such that $D^2 f(x) \underset{\sim}{<} B,$, in the sense that *B-D²f(x)* is positive semi-definite for all *x*. Then clearly

$$g(x,y) = f(y) + (x-y)' Df(y) + \frac{1}{2}(x-y)' B (x-y)$$

is a majorization function for *f.*

By defining the current *target*,

$$z = y - B^{-1} Df(y),$$

and by completing the square, we see that

$$g(x,y) = f(y) + \frac{1}{2}(x-z)' B (x-z) - \frac{1}{2} Df(y)' B^{-1} Df(y)$$

Thus step *k* of the majorization algorithm solves the least squares problem

$$\min_{x \in X} \left(x - z^{(k)}\right)' B \left(x - z^{(k)}\right).$$

We can choose the matrix *B* to be scalar, for instance by using an upper bound for the largest eigenvalue of *D²f(ξ)*. In that case computing the target simplifies, and all majorization subproblems are unweighted least squares problems.

In the case in which *X* is all of $\Re^n$ the quadratic majorization algorithm simply becomes

$$x^{(k+1)} = x^k - B^{-1} Df\left(x^{(k)}\right).$$

This algorithm will in general have a linear convergence rate $1-\lambda(x)$, where $\lambda(x)$ is the smallest eigenvalue of $B^{-1}D^2f(x)$ and x is the fixed point. A smaller B will give a more rapid convergence rate, but in general we cannot expect to see anything faster than linear convergence. If our bound B is really bad, then we may see very slow linear convergence.

6. Discussion

Majorization algorithms replace a complicated optimization problem by a sequence of simpler ones. In fact typically the subproblems are chosen in such a way that they are really simple to solve, and this is exactly what makes the algorithm attractive for really large problems such as the ones in tomography and microarray analysis. The quantities needed in an iterative step do not have a large footprint and can often be computed in parallel or from a stream of data. Convergence can be slow, and techniques to accelerate convergence may be necessary. But is often far more convenient to let a simple globally convergent ad-hoc algorithm run for hours than to try to fit huge and ill-conditioned matrices into memory in order to apply some suitably safe-guarded version of Newton's method.

References

1. J.De Leeuw. Quadratic and Cubic Majorization. Pre print series, UCLA Department of Statistics, 2006.
2. J. De Leeuw. Block Relaxation Methods in Statistics. In H.H. Bock, W.Lenski, and M.M. Richter, editors. *Information Systems and Data Analysis,* Berlin, 1994, Springer Verlag.
3. P. Hartman. On Functions Representable as a Difference of Two Convex Functions. *Pacific Journal of Mathematics*, 9:707-713,1959.
4. W.J.Heiser. Convergent Computing by Iterative Majorization: Theory and Applications in Multidimensional Data Analysis. In W.J. Krzanowski, editor, *Recent Advances in Descriptive Multivariate Analysis*, pages 157-189, Oxford: Clarendon Press, 1995.
5. K. Lange, D.R. Hunter, and I. Yang. Optimization Transfer Using Surrogate Objective Functions. *Journal of Computational and Graphical Statistics*, 9:1-20,2000.
6. J.M.Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables. Academic Press*, New York, N.Y., 1970.

Seagulls outside the February 2006 Program Chair Meeting in Alexandria, VA. (Photo courtesy of Juergen Symanzik)

"R has sounded the death knell for statistical computing research." Anonymous