MAJORIZING A MULTIVARIATE POLYNOMIAL OVER THE UNIT SPHERE

JAN DE LEEUW

1. PROBLEM

The problem studied in this note is to minimize a polynomial $P : \mathbb{R}^m \Rightarrow \mathbb{R}$ over the unit sphere $\mathbb{S} = \{x \mid x'x = 1\}$. Clearly the problem is well-defined, because the minimum always exists.

We use the standard notation $P(x) = \sum_\alpha p_\alpha x^\alpha$ for multivariate polynomials, where α are vectors of m integers, and

$$x^\alpha = \prod_{j=1}^m x_j^{\alpha_j}.$$

One important application we have in mind is minimizing polynomial functions of Jacobi plane rotations [De Leeuw, 2008], a problem that occurs in factor analysis, component analysis, multiway decomposition, and multidimensional scaling. Instead of using $\sin(\theta)$ and $\cos(\theta)$ which define the Jacobi rotation, we parametrize using x_1 and x_2 satisfying $x_1^2 + x_2^2 = 1$. This means using a bivariate polynomial on the sphere instead of a univariate trigoniometric polynomial.

Date: Friday 17th June, 2011 — 8h 52min — Typeset in TIMES ROMAN.

2000 *Mathematics Subject Classification.* 49M20.

Key words and phrases. Majorization, Polynomial Optimization.

2. ALGORITHM

We use quadratic majorization [Böhning and Lindsay, 1988; De Leeuw and Lange, 2009]. This requires us to find an upper bound for the quadratic term in the Taylor expansion of P. So, using g and H for the gradient and Hessian,

$$P(x)=P(y)+(x-y)'g(y)+\frac{1}{2}(x-y)'H(z)(x-y), \tag{1}$$

where z is on the line between x and y. Now

$$(x-y)'H(z)(x-y)\leq\|H(z)\|(x-y)'(x-y) \tag{2}$$

for any matrix norm $\|H(z)\|$.

If we use, for example, the ℓ_1 norm in (2)

$$\|H(z)\|=\sum_{i=1}^m\sum_{j=1}^m|h_{ij}(z)|=\sum_{i=1}^m\sum_{j=1}^m|\sum_\alpha p_{\alpha ij}z^\alpha|\leq\sum_{i=1}^m\sum_{j=1}^m\sum_\alpha|p_{\alpha ij}|, \tag{3}$$

because if x and y are in $\mathbb{S}$ we have z in the sphere, and thus $|z^\alpha|\leq 1$.

It follows that the function

$$Q(x\mid y)=P(y)+(x-y)'g(y)+\frac{1}{2}K(x-y)'(x-y),$$

with $K=\sum_{i=1}^m\sum_{j=1}^m\sum_\alpha|p_{\alpha ij}|$, is a majorization of P at y, and a step of the majorization algorithm minimizes $Q(x\mid y)$ over $x\in\mathbb{S}$.

The minimum is attained at

$$\hat{x}=\frac{y-\frac{1}{K}g(y)}{\|y-\frac{1}{K}g(y)\|},$$

and thus the majorization algorithm is the fixed step projected gradient algorithm

$$x^{(k+1)}=\frac{x^{(k)}-\frac{1}{K}g(x^{(k)})}{\|x^{(k)}-\frac{1}{K}g(x^{(k)})\|}. \tag{4}$$

3. EXAMPLE

Our polynomial is

$$P(x_1,x_2) = 3+4x_1-5x_2+x_1^2x_2^2-4x_1^3.$$

It has gradient

$$\begin{aligned} g_1(x_1,x_2) &= 4+2x_1x_2^2-12x_1^2,\\ g_2(x_1,x_2) &= -5+2x_1^2x_2, \end{aligned}$$

and Hessian

$$\begin{aligned} h_{11}(x_1,x_2) &= 2x_2^2-24x_1,\\ h_{12}(x_1,x_2) &= 4x_1x_2,\\ h_{21}(x_1,x_2) &= 4x_1x_2,\\ h_{22}(x_1,x_2) &= 2x_1^2. \end{aligned}$$

It follows that $K = 36$. Note that the maximum spectral norm over the sphere is 24, the maximum ℓ_1 norm is about 27.19041.

By making the substitution $x = \sin(\theta)$ and $y = \cos(\theta)$ the function becomes a trigonometric polynomial of a single variable θ. Computing it at 10,000 equally spaced values between -2π and $+2\pi$ show the minimum is about -2.805344 and occurs both at $\theta = 2.774618$ and $2.774618-2\pi = -0.3669747$.

The iterative majorization algorithm (4), started at $x = 1$ and $y = 0$, converges in 55 iterations to the function value -2.805344, attained at $x = -0.3588192$ and $y = 0.9334071$. Note that indeed $\arcsin(-0.3588192) = -0.3670025$, the approximate minimizer in the θ parametrization. We find the same solution if we start at $x = y = \frac{1}{2}\sqrt{2}$.

In the Appendix we give some R code that implements the majorization algorithm using the `multipol` package [Hankin, 2009]. It uses the same example. We could also use

```
1 p<-as.multipol(array(1:64,c(4,4,4))
```

For that example the algorithm converges smoothly, albeit slowly, to a minimum of -47.1303347324 at $x = 0.3340020, y = 0.3194996$, and $z = -0.8867709$. This requires 4827 iterations (with $\varepsilon = 1e-10$).

4. DISCUSSION

Clearly the same approach to algorithm construction can be used when majorizing a general twice-differentiable function on a sphere, as long as we can easily calculate upper bounds for the elements of the Hessian. By making the sphere large enough we can also tackle the problem of minimizing functions with continuous but not necessarily bounded derivatives.

REFERENCES

D. Böhning and B.G. Lindsay. Monotonicity of Quadratic-approximation Algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4): 641–663, 1988.

J. De Leeuw. Using Jacobi Plane Rotations in R. Preprint Series 556, UCLA Department of Statistics, Los Angeles, CA, 2008. URL `http://preprints.stat.ucla.edu/556/jacobi.pdf`.

J. De Leeuw and K. Lange. Sharp Quadratic Majorization in One Dimension. *Computational Statistics and Data Analysis*, 53:2471–2484, 2009.

R.K.S. Hankin. *multipol: multivariate polynomials*, 2009. URL `http://CRAN.R-project.org/package=multipol`. R package version 1.0-4.

APPENDIX A. CODE

```
library(multipol)

p<-as.multipol(matrix(c(3,4,0,-4,-5,0,0,0,0,0,1,0),4,3))

majPol <-function (p, xold, itmax = 100, eps = 1e-10, verbose = TRUE) {
    if (!is.multipol (p)) {
        p <- as.multipol (p)
    }
    r <- length (dim (p))
    s <- 1 : r
    g <- lapply (s, function (i) deriv (p, i))
    h <- lapply(g, function(f) lapply (s, function(i) deriv(f,i)))
    K <- sum (abs (unlist (h)))
    fold <- as.function (p) (xold)
    itel <- 1
    repeat {
        grad <- sapply(s, function (i) as.function (g [[i]]) (xold))
        xraw <- xold - grad / K
        xnew <- xraw / sqrt (sum (xraw ^ 2))
        fnew <- as.function (p) (xnew)
        if (verbose) {
            cat("Iteration: ",
                formatC(itel,digits=6,width=6),
                "   f old :   ",
                formatC(fold,digits=10,width=15,format="f"),
                "   f new :   ",
                formatC(fnew,digits=10,width=15,format="f"),
                "\n")
        }
        if ((itel == itmax) || ((fold - fnew) < eps)) {
            return (list (itel = itel, f = fnew, x = xnew))
        }
        itel <- itel + 1
        fold <- fnew
        xold <- xnew
    }
}
```

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90095-1554

E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

URL, Jan de Leeuw: http://gifi.stat.ucla.edu