Title
Majorizing a Multivariate Polynomial Over the Unit Sphere

Permalink
https://escholarship.org/uc/item/9867r6w9

Author
Leeuw, Jan de

Publication Date
2011-02-06

Peer reviewed
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 \( S = \{ x \mid x'x = 1 \} \). 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 \( \alpha \) 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 trigonometric polynomial.
2. ALGORITHM

We use quadratic majorization \cite{BoehningLindsay1988,DeLeeuwLange2009}. 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,

\begin{equation}
P(x) = P(y) + (x - y)'g(y) + \frac{1}{2}(x - y)'H(z)(x - y),
\end{equation}

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

\begin{equation}
(x - y)'H(z)(x - y) \leq \|H(z)\|(x - y)'(x - y)
\end{equation}

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

If we use, for example, the $\ell_1$ norm in \eqref{eq:2}

\begin{equation}
\|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}|,
\end{equation}

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

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

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

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

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

\begin{equation}
x^{(k+1)} = \frac{x^{(k)} - \frac{1}{K}g(x^{(k)})}{\|x^{(k)} - \frac{1}{K}g(x^{(k)})\|}.
\end{equation}
3. Example

Our polynomial is

\[ P(x_1, x_2) = 3 + 4x_1 - 5x_2 + x_1^2 x_2^2 - 4x_1^3. \]

It has gradient

\[
\begin{align*}
g_1(x_1, x_2) &= 4 + 2x_1x_2^2 - 12x_1^2, \\
g_2(x_1, x_2) &= -5 + 2x_1^2 x_2,
\end{align*}
\]

and Hessian

\[
\begin{align*}
h_{11}(x_1, x_2) &= 2x_2^2 - 24x_1, \\
h_{12}(x_1, x_2) &= 4x_1 x_2, \\
h_{21}(x_1, x_2) &= 4x_1 x_2, \\
h_{22}(x_1, x_2) &= 2x_1^2.
\end{align*}
\]

It follows that \( K = 36. \) Note that the maximum spectral norm over the sphere is 24, the maximum \( \ell_1 \) 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 \( \theta \). Computing it at 10,000 equally spaced values between \(-2\pi\) 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 \( \theta \) 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 \([\text{Hankin} 2009]\). It uses the same example. We could also use

```r
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


```r
library(multipol)

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