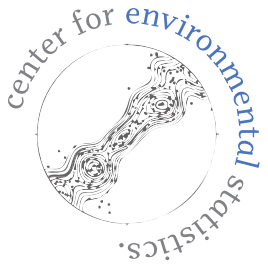

MATRIX NORMAL BLOCK EM: THE R PACKAGE MNBEM

JAN DE LEEUW AND WEI TAN TSAI

ABSTRACT. The Expectation-Maximization algorithm is derived as a special case of the Majorization Method. We specialize this general derivation to multivariate normal distributions, emphasizing in particular direct sum and direct product structures for the dispersion matrix. A special case inportant in environmental stastistics is missing data imputation in generalized growth curve models for the matrix variate normal distribution. The corresponding algorithms, with R code, are also given.

1. INTRODUCTION

The *majorization method* is a general approach, or family of approaches, to construct optimization methods. Some general publications about majorization are Kiers [1990]; De Leeuw [1994]; Heiser [1995]; Lange et al. [2000]; Hunter and Lange [2004]; De Leeuw and Lange [2009].

Suppose the problem is to minimize $f : \mathscr{X} \Rightarrow \mathbb{R}$ over $\mathscr{X} \subseteq \mathbb{R}^n$. A function $F : \mathscr{X} \otimes \mathscr{X} \Rightarrow \mathbb{R}$ is a *majorization function* if $f(x) \leq F(x,y)$ for all $x,y \in \mathscr{X}$ and $f(x) = F(x,x)$ for all $x \in \mathscr{X}$.

Date: Sunday 13th September, 2009 — 15h 6min — Typeset in TIMES ROMAN.

The iterative *majorization algorithm* finds the update of $x^{(k)}$ by computing

$$\mathscr{X}^{(k)} \triangleq \underset{x\in\mathscr{X}}{\textbf{argmax}}\, F(x,x^{(k)}).$$

If $x^{(k)} \in \mathscr{X}^{(k)}$ we stop. Else we select $x^{(k+1)} \in \mathscr{X}^{(k)}$. The *sandwich inequality*

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

shows that the algorithm either stops, or produces a decreasing sequence of function values. Under compactness and continuity conditions this implies convergence [Zangwill, 1969].

Of course if we are maximizing f, then we can construct a suitable minorization function and maximize that in each iterative step. To cover both minorization and majorization Lange et al. [2000] propose the name *MM algorithm*, where the first *M* stands for either majorization or minorization, and the second *M* stands for either maximation or minimization.

Majorization and minorization functions are usually derived from classical inequalities, from Taylor's Theorem, or from convexity considerations. The *Expectation-Maximization* or *EM algorithm* is a family of MM algorithms based on Jensen's Inequality, usually applied in the statistical context of computing maximum likelihood estimates [Dempster et al., 1977; McLachlan and Krishnan, 2008]. The general idea of using MM algorithms in data analysis came about by realizing that the EM algorithm, based on Jensen's Inequality, and the `SMACOF` method for multidimensional scaling [De Leeuw, 1977], based on the Cauchy-Schwartz Inequality, were both examples of a more general approach to algorithm construction.

1.1. **EM as MM.** Suppose that $g : X \otimes Y \Rightarrow \mathbb{R}^+$, where $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$. Define $f : X \to \mathbb{R}^+$ by

$$f(x) \triangleq \log \int_Y g(x,y)dy.$$

The problem we study in this paper is maximization of f over X.

Suppose $x,\tilde{x} \in X$. We assume that if $x \neq \tilde{x}$ then $g(x,y) \neq g(\tilde{x},y)$ for all $y \in Y$. Now

$$f(x) - f(\tilde{x}) = \log \frac{\int_Y g(x,y)dy}{\int_Y g(\tilde{x},y)dy} = \log \frac{\int_Y g(\tilde{x},y)\frac{g(x,y)}{g(\tilde{x},y)}dy}{\int_Y g(\tilde{x},y)dy}.$$

Let

$$h(x,y) \triangleq \frac{g(x,y)}{\int_Y g(x,y)dy}.$$

Then $\int_Y h(x,y)dy = 1$ for all x and

$$f(x) - f(\tilde{x}) = \log \int_Y h(\tilde{x},y) \frac{g(x,y)}{g(\tilde{x},y)} dy.$$

Applying Jensen's Inequality to the right hand side gives

$$f(x) > f(\tilde{x}) + k(x,\tilde{x}) - k(\tilde{x},\tilde{x}),$$

where we use the abbreviation

$$k(x,\tilde{x}) \stackrel{\Delta}{=} \int_Y h(\tilde{x},y) \log g(x,y) dy.$$

The function $F(x,\tilde{x}) = f(\tilde{x}) + k(x,\tilde{x}) - k(\tilde{x},\tilde{x})$ is the required minorization function.

This leads to the MM algorithm in which

$$\mathscr{X}^{(k)} \stackrel{\Delta}{=} \underset{x \in X}{\textbf{argmax}}\ F(x,x^{(k)}) = \underset{x \in X}{\textbf{argmax}}\ k(x,x^{(k)}),$$

and $x^{(k+1)} \in \mathscr{X}^{(k)}$.

2. MULTINORMAL EM

2.1. **General.** In the case of a $p-$dimensional multinormal density the parameter space X is some set of mean vectors μ and covariance matrices Σ. We write

$$g(\mu,\Sigma,y) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\{-\frac{1}{2}(y-\mu)'\Sigma^{-1}(y-\mu)\}.$$

Thus

$$f(\mu,\Sigma) = \log \int_Y g(\mu,\Sigma,y) dy,$$

and

$$h(\mu,\Sigma,y) = \frac{g(\mu,\Sigma,y)}{\int_Y g(\mu,\Sigma,y) dy}.$$

The MM algorithm minimizes

$$\ell(\mu,\Sigma,\tilde{\mu},\tilde{\Sigma}) = \log|\Sigma| + \int_Y \tilde{h}(\tilde{\mu},\tilde{\Sigma},y)(y-\mu)'\Sigma^{-1}(y-\mu) dy. \tag{1}$$

If we let

$$\tilde{m} \stackrel{\Delta}{=} \int_Y h(\tilde{\mu},\tilde{\Sigma},y) y dy, \tag{2a}$$

and

$$\tilde{V} \stackrel{\Delta}{=} \int_Y h(\tilde{\mu},\tilde{\Sigma},y)(y-\tilde{m})(y-\tilde{m})' dy, \tag{2b}$$

then

$$\ell(\mu,\Sigma,\tilde{\mu},\tilde{\Sigma}) = \log|\Sigma| + \mathbf{tr}\,\Sigma^{-1}\tilde{V} + (\tilde{m}-\mu)'\Sigma^{-1}(\tilde{m}-\mu). \tag{3}$$

2.2. **Marginal Constraints.** Suppose there is a $z \in \mathbb{R}^p$ such that $Y = \{z\} \otimes \mathbb{R}^q$, with $p+q=m$. Integration is over the last q of the m coordinates of $y=(z,u)$, with the first p coordinates fixed at z. Thus

$$f(\mu,\Sigma,z) = \log \int_{\mathbb{R}^q} g(\mu,\Sigma,z,u)du.$$

In this case h is the conditional density of u given z. Thus

$$h(\mu,\Sigma,u,z) = \frac{1}{\sqrt{(2\pi)^q|\Sigma_{u|z}|}}\exp\{-\frac{1}{2}(u-\mu_{u|z})'\Sigma_{u|z}^{-1}(u-\mu_{u|z})\},$$

where

$$\mu_{u|z} = \mu_u + \Sigma_{uz}\Sigma_{zz}^{-1}(z-\mu_z), \tag{4a}$$

$$\Sigma_{u|z} = \Sigma_{uu} - \Sigma_{uz}\Sigma_{zz}^{-1}\Sigma_{zu}. \tag{4b}$$

It follows directly that

$$\tilde{m} = \begin{bmatrix} z \\ \tilde{\mu}_u + \tilde{\Sigma}_{uz}\tilde{\Sigma}_{zz}^{-1}(z-\tilde{\mu}_z) \end{bmatrix}, \tag{5a}$$

$$\tilde{V} = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{\Sigma}_{uu} - \tilde{\Sigma}_{uz}\tilde{\Sigma}_{zz}^{-1}\tilde{\Sigma}_{zu} \end{bmatrix}. \tag{5b}$$

From the computational point of view we can most easily compute $\mu_{u|z}$ and $\Sigma_{u|z}$ by applying the sweep operator, explained for example in Lange [1999, Chapter 7], to the joint dispersion matrix $\tilde{\Sigma}$, sweeping the elements corresponding with z.

2.3. **Direct Sums.** Suppose Σ has direct sum structure, i.e. there are n positive semi-definite matrices Σ_i, or order m_i, such that

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 & \cdots & 0 \\ 0 & \Sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_n \end{bmatrix}.$$

The corresponding partition of μ is

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}.$$

Then

$$f(\mu, \Sigma) = \sum_{i=1}^{n} \log \int_{Y_i} g(\mu_i, \Sigma_i, y) dy,$$

where $Y_i \subseteq \mathbb{R}^{m_i}$. Also

$$\ell(\mu, \Sigma, \tilde{\mu}, \tilde{\Sigma}) = \sum_{i=1}^{n} \left\{ \log |\Sigma_i| + \int_{Y_i} h_i(\tilde{\mu}_i, \tilde{\Sigma}_i, y)(y-\mu_i)' \Sigma_i^{-1} (y-\mu_i) dy \right\}, \tag{6a}$$

where

$$h_i(\tilde{\mu}_i, \tilde{\Sigma}_i, y) = \frac{g(\tilde{\mu}_i, \tilde{\Sigma}_i, y_i)}{\int_{Y_i} g(\tilde{\mu}_i, \tilde{\Sigma}_i, y) dy}.$$

Thus, with obvious indexing notation,

$$\ell(\mu, \Sigma, \tilde{\mu}, \tilde{\Sigma}) = \sum_{i=1}^{n} \left\{ \log |\Sigma_i| + \mathbf{tr}\, \Sigma_i^{-1} \tilde{V}_i + (\tilde{m}_i - \mu_i)' \Sigma_i^{-1} (\tilde{m}_i - \mu_i) \right\}. \tag{6b}$$

This is basically the same as n replications of Equation (3).

2.4. **Repeated Independent Trials.** The results from Section 2.3 simplify if all Σ_i are equal to, say, the same positive definite matrix Σ of order m [1]Define $\tilde{V} \stackrel{\Delta}{=} \frac{1}{n} \sum_{i=1}^{n} \tilde{V}_i$. Then

$$\frac{1}{n} \ell(\mu, \Sigma, \tilde{\mu}, \tilde{\Sigma}) = \log |\Sigma| + \mathbf{tr}\, \Sigma^{-1} \tilde{V} + \frac{1}{n} \sum_{i=1}^{n} (\tilde{m}_i - \mu_i)' \Sigma^{-1} (\tilde{m}_i - \mu_i). \tag{7a}$$

If we collect all $\tilde{m}_i$ in the $n \times m$ matrix $\tilde{M}$ and all μ_i in the $n \times m$ matrix Ξ, then

$$\frac{1}{n} \ell(\mu, \Sigma, \tilde{\mu}, \tilde{\Sigma}) = \log |\Sigma| + \mathbf{tr}\, \Sigma^{-1} \tilde{V} + \frac{1}{n} \mathbf{tr}\, (\tilde{M} - \Xi) \Sigma^{-1} (\tilde{M} - \Xi)'. \tag{7b}$$

Define $S(\Xi) \stackrel{\Delta}{=} \frac{1}{n} (\tilde{M} - \Xi)'(\tilde{M} - \Xi)$. Then

$$\frac{1}{n} \ell(\mu, \Sigma, \tilde{\mu}, \tilde{\Sigma}) = \log |\Sigma| + \mathbf{tr}\, \Sigma^{-1} \tilde{V} + \mathbf{tr}\, \Sigma^{-1} S(\Xi). \tag{7c}$$

[1]Observe we use Σ both for the $m \times m$ column-covariances, and for $I \otimes \Sigma = \underbrace{\Sigma \oplus \cdots \oplus \Sigma}_{\text{n times}}$. It will be obvious from the context which Σ we mean.

If the μ_i are also equal we define $\tilde{m}\overset{\Delta}{=}\frac{1}{n}\sum_{i=1}^n\tilde{m}_i$. We then find

$$\frac{1}{n}\ell(\mu,\Sigma,\tilde{\mu},\tilde{\Sigma})=\log|\Sigma|+\mathbf{tr}\ \Sigma^{-1}\tilde{V}+(\tilde{m}-\mu)'\Sigma^{-1}(\tilde{m}-\mu)+\mathbf{tr}\ \Sigma^{-1}S, \tag{7d}$$

where $S\overset{\Delta}{=}\frac{1}{n}\sum_{i=1}^n(\tilde{m}_i-\tilde{m})(\tilde{m}_i-\tilde{m})'$.

2.5. **Direct Product (Kronecker) Structure.** Now consider the more general case in which the dispersion matrix is the $nm\times nm$ matrix $\Gamma\otimes\Sigma$, where Γ is of order n and Σ is of order m. This defines the matrix variate normal distribution, discussed in detail in Gupta and Nagar [2000]. The repeated independent trials from Section 2.4 are the special case in which $\Gamma=I$. In the usual interpretation we have an $n\times m$ matrix valued random variable $\underline{Y}$, and there is a row covariance matrix Γ and a column covariance matrix Σ, which combine to the direct product covariance matrix $\Gamma\otimes\Sigma$ of all nm variables $\underline{y}_{ij}$. Note that assuming Kronecker structure reduces the number of parameters in the dispersions from $\frac{1}{2}nm(nm+1)$ to $\frac{1}{2}n(n+1)+\frac{1}{2}m(m+1)$.

In environmental statistics direct product covariance structures are used to approximate the impractically large covariance matrices of space time time or vector-valued time series. In that context, covariance matrices with direct product, or Kronecker, structure are often called separable. See, for example, Matsuda and Yajima [2004], Lu and Zimmerman [2005] or Mitchell et al. [2006]. In the context of growth curve models Kronecker product dispersion matrices have been studied by many authors. Some recent interesting publications are Srivastava et al. [2008a,b]. For repeated measures data, Kronecker product covariance structures have been proposed by Naik and Rao [2001] and Roy and Khattree [2005]. In most of the papers cited, however, emphasis is on estimation and testing, and not on actual computation.

The log-likelihood for the matrix variate normal is

$$\log g(\Xi,\Gamma,\Sigma,Y)=m\log\Gamma+n\log\Sigma+\mathbf{tr}\ \Gamma^{-1}(Y-\Xi)\Sigma^{-1}(Y-\Xi)', \tag{8}$$

and the majorization function for multinormal EM becomes

$$\begin{aligned}\ell(\Xi,\Gamma,\Sigma,\tilde{\Xi},\tilde{\Gamma},\tilde{\Sigma})&=\\&=m\log|\Gamma|+n\log|\Sigma|+\mathbf{tr}\ (\Gamma^{-1}\otimes\Sigma^{-1})\tilde{V}+\mathbf{tr}\ \Gamma^{-1}(\tilde{M}-\Xi)'\Sigma^{-1}(\tilde{M}-\Xi).\end{aligned} \tag{9}$$

Observe that now $\tilde{V}$ is of order nm, and it has a non-zero row and column for all missing elements of the $n\times m$ data matrix.

3. MULTINORMAL MAXIMUM LIKELIHOOD USING BLOCK RELAXATION

The block relaxation method for multinormal maximum likelihood estimation [Oberhofer and Kmenta, 1974; De Leeuw, 1994] is designed to minimize

$$\mathscr{D}(\mu,\Sigma)=\log g(\mu,\Sigma,x)=\log|\Sigma|+(x-\mu)'\Sigma^{-1}(x-\mu)$$

over $\mu\in\mathscr{M}\subseteq\mathbb{R}^m$ and $\Sigma\in\mathscr{P}\in\mathbb{R}^{m\times m}$. The algorithm alternates optimization of μ and Σ, and computes its updates using the rule

$$\mu^{(k+1)}\in\operatorname*{\mathbf{argmin}}_{\mu\in\mathscr{M}}(x-\mu)'[\Sigma^{(k)}]^{-1}(x-\mu), \tag{10a}$$

$$\Sigma^{(k+1)}\in\operatorname*{\mathbf{argmin}}_{\Sigma\in\mathscr{P}}\log|\Sigma|+\mathbf{tr}\,\Sigma^{-1}(x-\mu^{(k+1)})(x-\mu^{(k+1)})'. \tag{10b}$$

The update rules in (10) can be relaxed to define generalized block methods. It is not necessary to minimize in each of the two subproblems, it is enough to strictly decrease, as long as this decrease is done with continuous maps. So suppose we have two continuous maps $F:\mathbb{R}^m\otimes\mathbb{R}^{m\times m}\to\mathbb{R}^m$ and $G:\mathbb{R}^m\otimes\mathbb{R}^{m\times m}\to\mathbb{R}^{m\times m}$. Define $\mu^{(k+1)}=F(\mu^{(k)},\Sigma^{(k)})$ and $\Sigma^{(k+1)}=G(\mu^{(k+1)},\Sigma^{(k)})$. Suppose

$$(x-\mu^{(k+1)})'[\Sigma^{(k)}]^{-1}(x-\mu^{(k+1)})<(x-\mu^{(k)})'[\Sigma^{(k)}]^{-1}(x-\mu^{(k)}),$$

and

$$\begin{aligned}\log|\Sigma^{(k+1)}|+\mathbf{tr}\,[\Sigma^{(k+1)}]^{-1}(x-\mu^{(k+1)})(x-\mu^{(k+1)})'<\\<\log|\Sigma^{(k)}|+\mathbf{tr}\,[\Sigma^{(k)}]^{-1}(x-\mu^{(k+1)})(x-\mu^{(k+1)})'\end{aligned}$$

This still defines a convergent algorithm.

3.1. **Block EM.**

Using our EM results we can extend the block relaxation algorithm to minimizing

$$\mathscr{D}(\mu,\Sigma)=\log\int_Y g(\mu,\Sigma,y)dy$$

One possible sequence of steps is

$$m^{(k)}=\int_Y h(\mu^{(k)},\Sigma^{(k)},y)y\,dy, \tag{11a}$$

$$V^{(k)}=\int_Y h(\mu^{(k)},\Sigma^{(k)},y)(y-m^{(k)})(y-m^{(k)})'dy, \tag{11b}$$

$$\mu^{(k+1)}\in\operatorname*{\mathbf{argmin}}_{\mu\in\mathscr{M}}(m^{(k)}-\mu)'[\Sigma^{(k)}]^{-1}(m^{(k)}-\mu), \tag{11c}$$

$$\Sigma^{(k+1)}\in\operatorname*{\mathbf{argmin}}_{\Sigma\in\mathscr{P}}\log|\Sigma|+\mathbf{tr}\,\Sigma^{-1}\{V^{(k)}+(m^{(k)}-\mu^{(k+1)})(m^{(k)}-\mu^{(k+1)})'\}. \tag{11d}$$

There are many variations possible, however. We can alternate 11c and 11d a number of times, before we go back to 11a and 11b. We can do steps 11a and 11b after each step 11c and after each step 11d. And so on. In some cases it will even be possible to replace 11c and 11d by

$$(\mu^{(k+1)},\Sigma^{(k+1)})\in\underset{\mu\in\mathscr{M},\Sigma\in\mathscr{P}}{\textbf{argmin}}\ \log|\Sigma|+\textbf{tr}\ \Sigma^{-1}\{V^{(k)}+(m^{(k)}-\mu)(m^{(k)}-\mu)'\}.$$

For instance if there are no constraints on μ, i.e. $\mathscr{M}=\mathbb{R}^m$, then $\mu^{(k+1)}=m^{(k)}$, and

$$\Sigma^{(k+1)}\in\underset{\Sigma\in\mathscr{P}}{\textbf{argmin}}\log|\Sigma|+\textbf{tr}\ \Sigma^{-1}V^{(k)}.$$

3.2. **Kronecker Weights.** Minimizing (9) requires some extra thought. We will use a block algorithms with five substeps.

$$m^{(k)}=\textbf{vec}(M^{(k)})=\int_Y h(\mu^{(k)},\Gamma^{(k)},\Sigma^{(k)},y)\textbf{vec}(Y)dy, \tag{12a}$$

$$V^{(k)}=\int_Y h(\mu^{(k)},\Gamma^{(k)},\Sigma^{(k)},y)(\textbf{vec}(Y)-m^{(k)})(\textbf{vec}(Y)-m^{(k)})'dy, \tag{12b}$$

$$\Xi^{(k+1)}\in\underset{\Xi\in\mathscr{M}}{\textbf{argmin}}\,\textbf{tr}\ \{[\Gamma^{(k)}]^{-1}(M^{(k)}-\Xi)[\Sigma^{(k)}]^{-1}(M^{(k)}-\Xi)'\}, \tag{12c}$$

$$\Gamma^{(k+1)}\in\underset{\Gamma\in\mathscr{G}}{\textbf{argmin}}\,m\log|\Gamma|+\textbf{tr}\ (\Gamma^{-1}\otimes[\Omega^{(k)}]^{-1})V^{(k)}+\textbf{tr}\ \Gamma^{-1}\hat{\Gamma}^{(k)}, \tag{12d}$$

$$\Sigma^{(k+1)}\in\underset{\Sigma\in\mathscr{S}}{\textbf{argmin}}\,n\log|\Sigma|+\textbf{tr}\ ([\Gamma^{(k+1)}]^{-1}\otimes\Omega^{-1})V^{(k)}+\textbf{tr}\ \Sigma^{-1}\hat{\Sigma}^{(k)}, \tag{12e}$$

where

$$\hat{\Gamma}^{(k)}=(M^{(k)}-\Xi^{(k+1)})[\Sigma^{(k)}]^{-1}(M^{(k)}-\Xi^{(k+1)})', \tag{13a}$$

$$\hat{\Sigma}^{(k)}=(M^{(k)}-\Xi^{(k+1)})'[\Gamma^{(k+1)}]^{-1}(M^{(k)}-\Xi^{(k+1)}). \tag{13b}$$

The *piece de résistance* is the term $\textbf{tr}\ (\Gamma^{-1}\otimes\Omega^{-1})V^{(k)}$, which involves multiplication of two $nm\times nm$ matrices. Since we have applications in mind where nm could be of order 10^4, the matrices could grow unpleasantly large. We can use the Kronecker structure of $\Gamma^{-1}\otimes\Omega^{-1}$ and the sparseness of $V^{(k)}$ to get considerable savings in computation and storage.

If we are updating Σ in the block relaxation process we use

$$\textbf{tr}\ (\Gamma^{-1}\otimes\Sigma^{-1})V^{(k)}=\textbf{tr}\ \Sigma^{-1}\sum_{i=1}^n\sum_{k=1}^n\gamma^{ik}V_{ik}^{(k)},$$

where the γ^{ik} are the elements of Γ^{-1} and the $V_{ik}^{(k)}$ are the $m \times m$ submatrices of $V^{(k)}$ corresponding with row i and row k of Y. Note that element j,ℓ of $V_{ik}^{(k)}$ is nonzero if and only if both y_{ij} and $y_{k\ell}$ are missing.

In the same way if we are updating Γ we use

$$\mathbf{tr}\,(\Gamma^{-1} \otimes \Sigma^{-1})V^{(k)} = \mathbf{tr}\,\Gamma^{-1} \sum_{j=1}^{m}\sum_{\ell=1}^{m} \sigma^{j\ell} V_{j\ell}^{(k)},$$

where now the $V_{j\ell}^{(k)}$ are the $n \times n$ submatrices of $V^{(k)}$ corresponding with column j and column ℓ of Y.

It is probably a good idea to start the iterations by minimizing the log-likelihood (8) over Σ, Γ, and Ξ and over the missing elements of Y. This avoids the huge Kronecker products altogether and can be assumed to give a good initial estimate of the structural parameters. At some point, we then let the algorithm (12) take over.

REFERENCES

J. De Leeuw. Applications of Convex Analysis to Multidimensional Scaling. In J.R. Barra, F. Brodeau, G. Romier, and B. Van Cutsem, editors, *Recent developments in statistics*, pages 133–145, Amsterdam, The Netherlands, 1977. North Holland Publishing Company.

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.

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

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM algorithm (with Discussion). *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

A.K. Gupta and D.K. Nagar. *Matrix Variate Distributions*. Chapman & Hall/CRC, Boca Raton, Florida, 2000.

W.J. Heiser. Convergent Computing by Iterative Majorization: Theory and Applications in Multidimensional Data Analysis. In W.J. Krzanowski, editor, *Recent Advantages in Descriptive Multivariate Analysis*, pages 157–189. Clarendon Press, Oxford, 1995.

D.R. Hunter and K. Lange. A Tutorial on MM Algorithms. *American Statistician*, 58(30–37), 2004.

H. Kiers. Majorization as a Tool for Optimizing a Class of Matrix Functions. *Psychometrika*, 55:417–428, 1990.

K. Lange. *Numerical Analysis for Statisticians*. Springer, Berlin, Heidelberg, New York, 1999.

K. Lange, D.R. Hunter, and I. Yang. Optimization Transfer Using Surrogate Objective Functions. *Journal of Computational and Graphical Statistics*, 9:1–20, 2000.

N. Lu and D.L. Zimmerman. The Likelihood Ratio Test for a Separable Covariance Matrix. *Statistics and Probability Letters*, 73:449–457, 2005.

Y. Matsuda and Y. Yajima. On Testing for Separable Correlations of Multivariate Time Series. *Journal of Time Series Analysis*, 25:501–528, 2004.

G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, New York, second edition, 2008.

M.W. Mitchell, M.G. Genton, and ML. Gumpertz. A Likelihood Ratio Test for Seperability of Covariances. *Journal of Multivariate Analysis*, 97:1025–1043, 2006.

D.N. Naik and S.S. Rao. Analysis of Multivariate Repeated Measures Data with a Kronecker Product Structured Covariance Matrix. *Journal of Applied Statistics*, 28:91–105, 2001.

W. Oberhofer and J. Kmenta. A General Procedure for Obtaining Maximum Likelihood Estimates in Generalized Regression Models. *Econometrica*, 42:579–590, 1974.

A. Roy and R. Khattree. On Implementation of a Test for Kronecker Product Covariance Structure for Multivariate Repeated Measures Data. *Statistical Methodology*, 2:297–306, 2005.

M.S. Srivastava, T. von Rosen, and D. von Rosen. Models with a Kronecker Product Covariance Structure: Estimation and Testing. *Mathematical Methods of Statistics*, 17:357–370, 2008a.

M.S. Srivastava, T. von Rosen, and D. von Rosen. Estimation in General Multivariate Linear Models with Kronecker Product Covariance Structure. Research Report 2008:1, Center of Biostatistics, Swedish University of Agricultural Sciences, 2008b.

W. I. Zangwill. *Nonlinear Programming: a Unified Approach*. Prentice-Hall, Englewood-Cliffs, N.J., 1969.

APPENDIX A. CODE

```
#
#   mnem package
#   Copyright (C) 2009  Jan de Leeuw <deleeuw@stat.ucla.edu>
#   UCLA Department of Statistics, Box 951554, Los Angeles, CA 90095-1554
#
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2 of the License, or
#   (at your option) any later version.
#
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warrant(y) of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#   GNU General Public License for more details.
#
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#
######################################################################
#
# version 0.1, 2009-06-06   Initial  Release
# version 0.2, 2009-06-08   Added some stuff
#

# which(outer(i,j,function(x,y) is.na(x)&is.na(y)))

imputeK<-function(y,v,w,mu) {
n<-nrow(y); m<-ncol(y)
vsig<-kronecker(v,w); vy<-as.vector(t(y)); vmu<-as.vector(t(mu))
vim<-imputeY(vy,vsig,vmu)
return(list(vim=matrix(vim$yimp,n,m,byrow=TRUE),vvv=vim$vimp))
}

sigmaHat<-function(vmat,v) {
n<-nrow(v); m<-nrow(vmat)/n; mm<-1:m; shat<-matrix(0,m,m)
for (i in 1:n) for (k in 1:n)
    shat<-shat+v[i,k]*vmat[mm+(i-1)*m,mm+(k-1)*m]
return(shat)
}

gammaHat<-function(vmat,w) {
m<-nrow(w); n<-nrow(vmat)/m; mm<-(0:(n-1))*m; ghat<-matrix(0,n,n)
for (j in 1:m) for (l in 1:m)
    ghat<-ghat+w[j,l]*vmat[mm+j,mm+l]
return(ghat)
}

# given a multivariate normal with mean mu and
# covariance matrix sig, and a vector of observations y,
# impute the missing values in y and the corresponding
```

```
# conditional covariance matrix

imputeY<-function(y,sig,mu) {
n<-length(y); nm<-is.na(y)
indi<-which(!nm); v<-matrix(0,n,n)
cs<-condiStat(sig,mu,y[indi],indi)
y[nm]<-cs$cmean; v[nm,nm]<-cs$cdisp
return(list(yimp=y,vimp=v))
}

# given a multivariate normal with mean mu and
# covariance matrix sig, compute the conditional
# mean and variance if we fix the variables indexed
# with indi to be equal to z

condiStat<-function(sig,mu,z,indi) {
m<-nrow(sig); jndi<-(1:m)[-indi]
mv<-rep(0,m); mv[indi]<-mu[indi]-z; mv[jndi]<-mu[jndi]
aa<-sweeper(cbind(sig,mv),indi)
return(list(cmean=aa[jndi,m+1],cdisp=aa[jndi,jndi]))
}

# beaton's sweep function

sweeper<-function(a,indi) {
n<-nrow(a); m<-length(indi)
for (j in indi) {
    pv<-a[j,j]
    if (pv == 0) next()
    pr<-a[j,-j]; pc<-a[-j,j]
    a[j,j] <- -1/pv
    a[j,-j] <- pr/pv
    a[-j,j] <- pc/pv
    a[-j,-j] <- a[-j,-j]-outer(pc,pr)/pv
    }
return(a)
}

# this minimizes trace{(y-mu)'(y-mu)} over the
# missing elements of y

imputeOLS<-function(y,mu) return(ifelse(is.na(y),mu,y))

imputeWLS<-function(y,k,ginv,sinv,mu,eps=1e-6,itmax=100,verbose=FALSE)
{
n<-nrow(y); m<-ncol(y); itel<-1
res<-y-mu; ras<-ginv%*%res%*%sinv; ff<-sum(ras*res)
wgt<-outer(diag(ginv),diag(sinv))
repeat {
    thmax<-0
    for (i in 1:n) for (j in 1:m) {
            if (!is.na(k[i,j])) next()
            rij<-ras[i,j]; wij<-wgt[i,j]
```

```
105           th<-rij/wij
106           thmax<-max(thmax,abs(th))
107           z[i,j]<-z[i,j]+th
108           ff<-ff-(rij^2)/wij
109           ras<-ras+th*outer(ginv[,i],sinv[,j])
110           }
111       if (verbose)
112               cat("itel",formatC(itel,format="d",width=4)," maxth",formatC(thmax,
                      format="f",digits=8,width=15)," func",formatC(ff,format="f",digits
                      =8,width=15),"\n")
113       if ((thmax < eps) || (itel == itmax)) break()
114       itel<-itel+1
115       }
116   return(y)
117   }
```

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

E-mail address, Wei Tan Tsai: tsai@stat.ucla.edu

URL, Wei Tan Tsai: http://www.stat.ucla.edu/~tsai/