On a Multiplicative Algorithm for Computing Bayesian D-optimal Designs

Yaming Yu

Department of Statistics
University of California
Irvine, CA 92697, USA
yamingy@uci.edu

Abstract

We use the minorization-maximization principle (Lange, Hunter and Yang 2000) to establish the monotonicity of a multiplicative algorithm for computing Bayesian D-optimal designs. This proves a conjecture of Dette, Pepelyshev and Zhigljavsky (2008). We also study a nearest neighbor exchange strategy and show that it can dramatically improve the speed of the multiplicative algorithm.

Keywords: Bayesian D-optimality; experimental design; MM algorithms; monotonic convergence; overrelaxation.

1 Introduction

Multiplicative algorithms (Silvey, Titterington and Torsney 1978; Torsney and Mandal 2006; Dette, Pepelyshev and Zhigljavsky 2008) are often employed in numerical computation of optimal designs (approximate theory; see Kiefer 1974, Silvey 1980, and Pukelsheim 1993). These iterative algorithms are simple, easy to implement, and often increase the optimality criterion monotonically. In the case of D-optimality, for example, monotonicity of the algorithm of Silvey et al. (1978) is well known (Titterington 1976; Pázman 1986); see Yu (2010a) and the references therein for further results. Monotonicity is an important property as it implies convergence under mild conditions.

Bayesian D-optimality is a widely used design criterion that can accommodate prior uncertainty in the parameters (see Chaloner and Larntz 1989 and Chaloner and Verdinelli 1995).

Multiplicative algorithms extend naturally from D-optimality to Bayesian D-optimality (Dette et al. 2008). Although the form of the algorithms is just as simple as in the D-optimal case, a corresponding monotonicity result is still lacking. In the context of nonlinear regression, Dette et al. (2008) conjecture the monotonicity of a class of algorithms for computing Bayesian D-optimal designs. The main theoretical contribution of this work is to confirm their monotonicity conjecture.

Our technical devices include convexity and the minorization-maximization principle (MM; Lange, Hunter and Yang 2000; Hunter and Lange 2004). Similar ideas play a key role in settling the related Titterington's (1978) conjecture (see Yu 2010a, 2010b). Minorization-maximization (or bound optimization) is a general method for constructing iterative algorithms that increase an objective function $\phi(w)$ monotonically. We first construct a function $Q(w;\tilde{w})$ such that $\phi(w) \geq Q(w;\tilde{w})$ for all w and $\tilde{w}$, and $\phi(w) = Q(w;w)$. Suppose the current iterate is $w^{(t)}$. We choose $w^{(t+1)}$ to increase the Q function, i.e.,

$$Q\left(w^{(t+1)};w^{(t)}\right) \geq Q\left(w^{(t)};w^{(t)}\right). \tag{1}$$

Then $w^{(t+1)}$ also increases the objective function ϕ, because

$$\phi\left(w^{(t+1)}\right) \geq Q\left(w^{(t+1)};w^{(t)}\right) \geq Q\left(w^{(t)};w^{(t)}\right) = \phi\left(w^{(t)}\right).$$

The usual MM algorithm chooses $w^{(t+1)}$ to maximize $Q\left(\cdot;w^{(t)}\right)$. Since we only require (1), it is proper to call this strategy a *general MM algorithm*. The general MM algorithm is an extension of the general expectation-maximization algorithm (GEM; Dempster, Laird and Rubin 1977).

In Section 2 we state our monotonicity result and illustrate with a simple logistic regression example. Section 4 proves the monotonicity result. Specifically, the algorithm of Dette et al. (2008) for computing Bayesian D-optimal designs is derived as a general MM algorithm.

Multiplicative algorithms are known to be slow sometimes. This can be attributed to the difficulty of apportioning mass among adjacent design points. In Section 3, we consider a nearest neighbor exchange strategy and show that it can improve the speed considerably. Numerical examples include logistic regression and two other nonlinear models.

2 Theoretical Result and Illustration

We focus on a finite design space $\mathcal{X} = \{x_1, \ldots, x_n\}$. Let θ be the $m \times 1$ parameter of interest, and let $A_i(\theta)$ denote the $m \times m$ Fisher information matrix provided by a unit assigned to design point x_i. The so-called Bayesian D-optimality (Chaloner and Larntz 1989) seeks to maximize

$$\phi(w) \equiv \int \log\det M(w,\theta)\,\mathrm{d}\pi(\theta),$$

where $\pi(\theta)$ is a probability distribution representing prior knowledge about θ, and

$$M(w,\theta) = \sum_{i=1}^{n} w_i A_i(\theta).$$

This is an extension of local D-optimality which chooses the design weights w_i to maximize the log-determinant of the Fisher information for a fixed θ. It can also be viewed as a large sample approximation to Lindley's (1956) criterion based on Shannon information. Here $w = (w_1, \ldots, w_n) \in \bar{\Omega}$, and $\bar{\Omega}$ denotes the closure of $\Omega = \{w : \sum_{i=1}^{n} w_i = 1,\ w_i > 0\}$. To convert w to a finite-sample design, some rounding procedure is needed (Pukelsheim 1993, Chapter 12). The matrices $A_i(\theta)$ are assumed to be well defined and nonnegative definite for every θ.

Let us consider the following algorithm for maximizing $\phi(w)$. Define

$$d_i(w) = \int tr(M^{-1}(w,\theta)A_i(\theta))\,\mathrm{d}\pi(\theta).$$

Algorithm I

Set $w^{(0)} = (w_1^{(0)}, \ldots, w_n^{(0)}) \in \Omega$. That is, $w_i^{(0)} > 0$ for all i.

For $t = 0, 1, \ldots$, compute

$$w_i^{(t+1)} = w_i^{(t)} \frac{d_i\left(w^{(t)}\right) - \alpha^{(t)}}{m - \alpha^{(t)}}, \quad i = 1, \ldots, n, \tag{2}$$

where $\alpha^{(t)}$ satisfies

$$\alpha^{(t)} \le \frac{1}{2} \min_{i=1}^{n} d_i\left(w^{(t)}\right). \tag{3}$$

Iterate until convergence.

A commonly used convergence criterion is

$$\max_{i=1}^{n} d_i\left(w^{(t)}\right) \le m + \epsilon, \tag{4}$$

where ϵ is a small positive constant. This is based on the general equivalence theorem (Kiefer and Wolfowitz 1960; Whittle 1973), which characterizes any maximizer of $\phi(w)$, $\hat{w}$, by $\max_{i=1}^{n} d_i(\hat{w}) = m$.

Algorithm I slightly generalizes the one proposed by Dette et al. (2008). In a regression context, Dette et al. (2008) prove that Algorithm I is monotonic for D-optimality, i.e., when $\pi(\theta)$ is a point mass. Numerical examples support the conjecture that Algorithm I is monotonic for Bayesian D-optimality in general. We shall confirm this conjecture (Theorem 1).

Theorem 1. *Assume $\phi(w)$ is finite for at least one $w \in \Omega$. Let $w^{(t)}, w^{(t+1)} \in \Omega$ satisfy (2) and (3). Then we have*

$$\phi\left(w^{(t+1)}\right) \geq \phi\left(w^{(t)}\right),$$

with equality only if $w^{(t+1)} = w^{(t)}$.

Once strictly monotonicity is established, global convergence holds under mild conditions. We state such a result where $\alpha^{(t)}$ takes a convenient parametric form.

Theorem 2. *Assume $\phi(w)$ is finite for at least one $w \in \Omega$. Let $w^{(t)}$ be a sequence generated by (2), starting with $w^{(0)} \in \Omega$. Assume*

$$\alpha^{(t)} = \frac{a}{2} \min_{i=1}^{n} d_i\left(w^{(t)}\right), \tag{5}$$

where $a \in [0, 1]$ is a constant. Then all limit points of $w^{(t)}$ are global maxima of $\phi(w)$ on $\bar{\Omega}$.

Note that a limit point of $w^{(t)}$ may have some zero coordinates, although we require the starting value $w_i^{(0)} > 0$ for all i. Also, $\alpha^{(t)}$ changes from iteration to iteration. Nevertheless, based on Theorem 1, Theorem 2 can be established by an argument similar to that of Theorem 2 of Yu (2010a) (details omitted).

A natural question is the choice of $\alpha^{(t)}$. Given similar computing costs per iteration, it is reasonable to choose $\alpha^{(t)}$ based on the convergence rate. For D-optimal designs, i.e., when $\pi(\theta)$ is a point mass, Yu (2010b) analyzes the convergence rate of Algorithm I. We expect the results to carry over to Bayesian D-optimality. Specifically, treating the $\alpha^{(t)} \equiv 0$ case as the basic algorithm, we can view Algorithm I with $\alpha^{(t)} > 0$ as an overrelaxed version (in the sense of successive overrelaxation in numerical analysis; see Young 1971). See Yu (2010d) for further work on such overrelaxed algorithms that maintain monotonicity. At each iteration, overrelaxation

multiplies the step length of the basic algorithm by $m/(m-\alpha^{(t)})$. Noting $\sum_{i=1}^{n} d_i(w)w_i = m$ and (3), we have $\alpha^{(t)} \leq m/2$. That is, $m/(m-\alpha^{(t)}) \leq 2$. Thus, roughly speaking, overrelaxation can at most double the speed of the basic algorithm. A caveat is that, when the basic algorithm is very fast, overrelaxation may slow it down due to overshooting. Nevertheless, the examples provided by Dette et al. (2008) indicate that this rarely happens in practice. The slowness of the basic algorithm is usually the main concern.

For the rest of this section we illustrate our theoretical results with a logistic regression model

$$\Pr(y=1|x,\theta) = 1 - \Pr(y=0|x,\theta) = \left(1+\exp\left(-x^\top\theta\right)\right)^{-1}.$$

More examples can be found in Dette et al. (2008). Consider the design space

$$\mathcal{X}_1 = \left\{x_i = (1, i/10-1)^\top : \ i = 1,\ldots,30\right\}.$$

The Fisher information for θ from a unit assigned to x_i is

$$A_i(\theta) = x_i x_i^\top \frac{\exp(\eta_i)}{(1+\exp(\eta_i))^2}, \quad \eta_i \equiv x_i^\top\theta.$$

Suppose the distribution $\pi(\theta)$ assigns probability $1/25$ to each point in the following set

$$\left\{(i,j)^\top : \ i,j = -2,-1,0,1,2\right\}.$$

We implement Algorithm I to compute the Bayesian D-optimal design. The $\alpha^{(t)}$ is specified by (5) with several choices of a. Each algorithm is started at the uniform design $w^{(0)} = (1/30,\ldots,1/30)$, and we consider two convergence criteria corresponding to (4) with $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$ respectively. Table 1, which records the iteration counts, shows the advantage of using larger a ($a \leq 1$). The large iteration counts when $\epsilon = 10^{-4}$ illustrate the potential slow convergence of Algorithm I. We also display the optimality criterion $\phi\left(w^{(t)}\right)$ in Figure 1. As Theorem 1 claims, $\phi\left(w^{(t)}\right)$ increases monotonically for each algorithm.

Table 2 records the design weights as calculated by Algorithm I with $a = 1$. Note that, as the more stringent criterion $\epsilon = 10^{-4}$ is adopted, the weights assigned to the middle cluster x_i, $i = 14,\ldots,18$, become more concentrated around x_{16}. One interpretation is that Algorithm I sometimes has difficulty apportioning mass among adjacent design points, and therefore the convergence is slow. This also hints at potential remedies for the slow convergence. For computing D-optimal designs, Yu (2010c) proposes a "cocktail algorithm" that combines three different

Table 1: Iteration counts for Algorithm I with $\alpha^{(t)}$ specified by (5).

	$a = 0$	$a = 1/4$	$a = 1/2$	$a = 3/4$	$a = 1$
$\epsilon = 10^{-3}$	929	823	718	613	507
$\epsilon = 10^{-4}$	4112	3643	3175	2706	2238

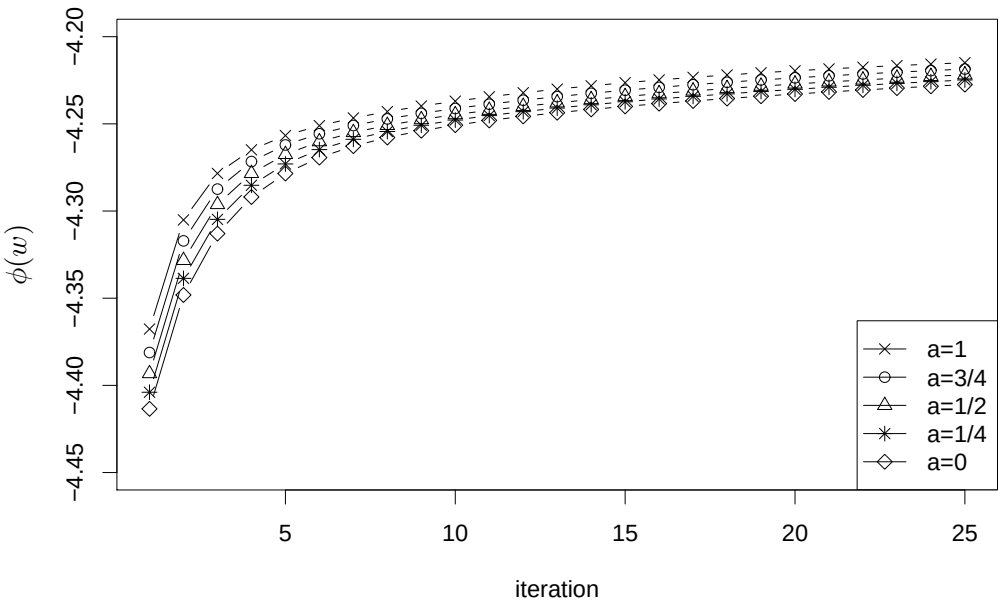

Figure 1: Monotonicity of $\phi\left(w^{(t)}\right)$ for Algorithm I.

Table 2: Output (design weights) of Algorithm I with $a = 1$ and two convergence criteria.

design points	x_1	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{30}
output ($\epsilon = 10^{-3}$)	0.434	0.006	0.073	0.114	0.035	0.003	0.334
output ($\epsilon = 10^{-4}$)	0.435	0.000	0.026	0.204	0.002	0.000	0.334

strategies for fast monotonic convergence. One of the ingredients, a special case of Algorithm I, is a multiplicative algorithm (Silvey et al. 1978). Another ingredient is a strategy that facilitates mass exchange between adjacent design points. In Section 3 we extend the cocktail algorithm to Bayesian D-optimality and show its effectiveness empirically.

3 Nearest Neighbor Exchanges to Improve Algorithm I

As mentioned in Section 2, one strategy to improve Algorithm I is to add steps that exchange the mass between adjacent support points. Specifically, suppose the design points $x_1, \ldots, x_n$ are ordered so that whenever $|i - j|$ is small, the information matrices $A_i(\theta)$ and $A_j(\theta)$ are close as points in a suitable metric space. Often a natural ordering is available, as in the logistic regression example of Section 2. Let $x_{i_1}, \ldots, x_{i_{p+1}}$ be the support points of the current iteration $w^{(t)}$. We consider exchanging the mass between x_{i_k} and $x_{i_{k+1}}$ for $k = 1, \ldots, p$ in turn, i.e.,

$$w^{(t+k/p)} = VE\left(i_k, i_{k+1}, w^{(t+(k-1)/p)}\right), \quad k = 1, \ldots, p.$$

The same strategy can be used to improve the EM algorithm for maximum likelihood estimation of mixture proportions (Yu, 2010e). Here $\tilde{w} = VE(j, l, w),\ j \neq l$, denotes an exchange of mass between x_j and x_l, i.e.,

$$\tilde{w}_i = \begin{cases} w_i, & i \notin \{j, l\}, \\ w_i + \delta, & i = j, \\ w_i - \delta, & i = l, \end{cases} \tag{6}$$

where $\delta \in [-w_j, w_l]$ is chosen so that $\phi(\tilde{w}) \geq \phi(w)$. Unlike in the D-optimal case, the step length δ that maximizes $\phi(\tilde{w})$ is usually not in closed form. However, Newton's method can handle this one-dimensional convex optimization problem. We implement (6) by performing one iteration of Newton's method, constraining δ to $[-w_j, w_l]$, and then halving δ (if necessary)

until $\delta \partial \phi(\tilde{w}) / \partial \delta \geq 0$. Concavity then guarantees $\phi(\tilde{w}) \geq \phi(w)$. This simple method seems to be both fast and stable in our experience.

Our nearest neighbor exchange strategy can potentially transfer all the mass from one w_i to its neighbor. This happens when the Newton iteration overshoots and δ is set at either $-w_j$ or w_l in (6). While reducing the support set can be beneficial, it may also accidentally eliminate a necessary support point, and prevent the algorithm from converging to a global maximum. As in Yu (2010c, 2010e), an easy remedy for this problem is to add one more step based on the vertex direction method (VDM; Fedorov 1972). Given the current w, we first find an index $i^{\#}$ such that $d_{i^{\#}}(w) = \max_{1 \leq i \leq n} d_i(w)$. Then we update w to $\tilde{w}$ with

$$\tilde{w}_i = \begin{cases} (1-\delta) w_i, & i \neq i^{\#}, \\ (1-\delta) w_i + \delta, & i = i^{\#}, \end{cases} \tag{7}$$

where $\delta \in [0, 1]$ is chosen so that $\phi(\tilde{w}) \geq \phi(w)$. The same Newton iteration followed by step-halving used in (6) is employed here.

The "cocktail algorithm," based on VDM, nearest neighbor exchanges, and Algorithm I, can be summarized as follows.

Algorithm II

Choose $w^{(0)} \in \bar{\Omega}$ such that $\phi\left(w^{(0)}\right) > -\infty$.

At iteration t, perform a VDM step (7), then the set of nearest neighbor exchanges, and then an iteration of (2), i.e., the multiplicative algorithm.

Algorithm II maintains the monotonicity of ϕ because each sub-step does so. Although more complicated than Algorithm I, Algorithm II is still easy to implement. Our computer code in R has fewer than 150 lines and is available upon request from the author. To illustrate the improvement in speed, we compare Algorithms I and II according to both the number of iterations and the computing time as measured by the R function system.time(). An iteration of the cocktail algorithm includes one iteration of VDM, the nearest neighbor exchanges, and an iteration of (2). The computing cost per iteration, however, does not increase by much, because the nearest neighbor exchanges and the iteration (2) only work with the current support set of $w^{(t)}$. We shall focus on the computing time comparisons.

We consider the logistic regression example of Section 2, with design spaces

$$\mathcal{X}_j = \left\{ x_i = (1, i/(10j) - 1)^\top : \ i = 1, \ldots, 30j \right\}, \quad j = 1, 2, 3.$$

The same $\pi(\theta)$ as in Section 2 is assumed. We also consider a Michaelis-Menten-type model

$$y|(x, \theta) \sim \theta_1 + \frac{\theta_3 x}{\theta_2 + x} + \mathrm{N}(0, \sigma^2),$$

and an exponential regression model

$$y|(x, \theta) \sim \theta_1 + \theta_3 \exp(-\theta_2 x) + \mathrm{N}(0, \sigma^2),$$

with design spaces

$$\mathcal{X}_{j+3} = \{ x_i = i/(10j) : \ i = 1, \ldots, 30j \}, \quad j = 1, 2, 3.$$

In these two models $\theta = (\theta_1, \theta_2, \theta_3)^\top$ and we assume that $\pi(\theta)$ is uniform with respect to θ_2 over the set $\{i/5 : \ i = 1, \ldots, 10\}$. The distribution on (θ_1, θ_3) is immaterial because of linearity.

As in Yu (2010c), we start the cocktail algorithm at the uniform design over a set of approximately $2m$ randomly sampled support points. Algorithm I is started at the uniform design over all n points. For Algorithm I, we set the overrelaxation parameter $a = 1$ in (5). Algorithm II, which is very fast, does not seem to benefit from using $a > 0$, and we set $a = 0$.

Table 3 (for logistic regression), Table 4 (for the Michaelis-Menten-type model) and Table 5 (for the exponential regression model) record the number of iterations and computing time for the two algorithms according to the convergence criterion (4) with $\epsilon = 10^{-4}$. For each model Algorithm II is a dramatic improvement. It reduces the computing time of Algorithm I by large factors, the reduction being more significant when the design space is more finely discretized. Replications lead to the same conclusion. Overall the comparison is very similar to that of the D-optimal case (Yu 2010c).

4 Monotonicity of Algorithm I

This section proves Theorem 1. We need Lemma 1, which slightly extends Lemma 1 of Dette et al. (2008).

Lemma 1. *For fixed θ, $\det M(w, \theta)$ is a polynomial in $w_1, \ldots, w_n$ with nonnegative coefficients.*

Table 3: Computing time (in seconds) and iteration counts in the logistic regression example.

	Computing time			Iteration count		
	$\mathcal{X}_1$	$\mathcal{X}_2$	$\mathcal{X}_3$	$\mathcal{X}_1$	$\mathcal{X}_2$	$\mathcal{X}_3$
Algorithm I	368.9	1544.0	2523.0	2238	4796	5279
Algorithm II	4.4	8.0	12.1	11	15	18

Table 4: Computing time (in seconds) and iteration counts in the Michaelis-Menten-type example.

	Computing time			Iteration count		
	$\mathcal{X}_4$	$\mathcal{X}_5$	$\mathcal{X}_6$	$\mathcal{X}_4$	$\mathcal{X}_5$	$\mathcal{X}_6$
Algorithm I	31.6	107.6	564.4	461	793	2758
Algorithm II	0.9	2.4	2.6	6	11	10

Table 5: Computing time (in seconds) and iteration counts in the exponential regression example.

	Computing time			Iteration count		
	$\mathcal{X}_4$	$\mathcal{X}_5$	$\mathcal{X}_6$	$\mathcal{X}_4$	$\mathcal{X}_5$	$\mathcal{X}_6$
Algorithm I	54.2	162.4	583.1	764	1269	2867
Algorithm II	2.0	1.8	2.4	12	9	9

Proof. Let I_m denote the $m \times m$ identity matrix, and define an $m \times (mn)$ matrix G by

$$G = (G_1, \ldots, G_{mn}) \equiv \left(A_1^{1/2}(\theta), \ldots, A_n^{1/2}(\theta) \right).$$

We have

$$M(w, \theta) = G(\mathrm{Diag}(w) \otimes I_m) G^\top.$$

The Cauchy-Binet formula (Horn and Johnson 1990, Chapter 0) yields

$$\det M(w, \theta) = \sum_{1 \le i_1 < \cdots < i_m \le mn} h(i_1, \ldots, i_m) u_{i_1} \cdots u_{i_m},$$

where $h(i_1, \ldots, i_m) = \det^2(G_{i_1}, \ldots, G_{i_m})$, and u_i denotes the ith diagonal of $\mathrm{Diag}(w) \otimes I_m$. The claim holds because u_i is equal to one of w_j, and $h(i_1, \ldots, i_m) \ge 0$. □

Lemma 2 serves as a building block for constructing our minorization-maximization strategy.

Lemma 2. *Let $g(w)$ be a nonzero polynomial in $w = (w_1, \ldots, w_n)$ with nonnegative coefficients. Define*

$$Q(w; \tilde{w}) = \sum_{i=1}^{n} \frac{\partial g(\tilde{w})}{\partial w_i} \frac{\tilde{w}_i}{g(\tilde{w})} \log w_i, \quad w, \tilde{w} \in \Omega.$$

Then we have

$$Q(w; \tilde{w}) - Q(\tilde{w}; \tilde{w}) \le \log g(w) - \log g(\tilde{w}).$$

Proof. Write $g(w) = \sum_{j=1}^{J} c_j f_j(w)$ where $c_j > 0$ and $f_j(w)$ are monomials in w. We have

$$\sum_{i=1}^{n} \frac{\partial f_j(\tilde{w})}{\partial w_i} \tilde{w}_i \log w_i = f_j(\tilde{w}) \log f_j(w), \quad j = 1, \ldots, J,$$

because f_j are monomials. Multiplying both sides by $c_j / g(\tilde{w})$ and then summing over j yield

$$Q(w; \tilde{w}) = \sum_{j=1}^{J} \frac{c_j f_j(\tilde{w})}{g(\tilde{w})} \log f_j(w).$$

Hence

$$\begin{aligned} Q(w; \tilde{w}) - Q(\tilde{w}, \tilde{w}) - \log g(w) + \log g(\tilde{w}) &= \sum_{j=1}^{J} \frac{c_j f_j(\tilde{w})}{g(\tilde{w})} \log \frac{c_j f_j(w)/g(w)}{c_j f_j(\tilde{w})/g(\tilde{w})} \\ &\le \log \left(\sum_{j=1}^{J} \frac{c_j f_j(w)}{g(w)} \right) \\ &= 0, \end{aligned}$$

where the inequality holds by Jensen's inequality applied to the concave function $\log x$. □

Lemma 3 is implicit in Dette et al. (2008); see also Yu (2010d). The proof is included for completeness.

Lemma 3. *Define* $Q(w) = \sum_{i=1}^n q_i \log w_i,\ q, w \in \Omega$. *For a fixed* w, *let* α *be a scalar that satisfies*

$$\alpha \le \frac{1}{2} \min_{i=1}^{n} \frac{q_i}{w_i}.$$

Then we have

$$Q(\hat{w}) \ge Q(w), \quad \hat{w} \equiv \frac{q - \alpha w}{1 - \alpha},$$

with equality only if $\hat{w} = w$.

Proof. Letting $r_i = q_i / w_i$, we have

$$\begin{aligned} Q(\hat{w}) - Q(w) &= \sum_{i=1}^{n} w_i r_i \log \frac{r_i - \alpha}{1 - \alpha} \\ &\ge \bar{r} \log \frac{\bar{r} - \alpha}{1 - \alpha} \\ &= 0, \end{aligned}$$

where $\bar{r} = \sum_i w_i r_i = 1$ (hence the last equality), and the inequality follows by Jensen's inequality applied to the function $x \log(x - \alpha)$, which is convex on $x \ge \max\{0,\, 2\alpha\}$. Hence $Q(\hat{w}) \ge Q(w)$. By strict convexity, equality holds only when $r_i = \bar{r} = 1$ for all i, i.e., when $\hat{w} = w$. □

Proof of Theorem 1. It is easy to see that, if $\phi(w)$ is finite for any $w \in \Omega$, then it is finite for all $w \in \Omega$. Define $g(w, \theta) = \det M(w, \theta)$. Because $\phi(w)$ is finite, we have $g(w, \theta) > 0$ almost surely with respect to $\pi(\theta)$. By Lemma 1, for fixed θ, $g(w, \theta)$ is a polynomial in w with nonnegative coefficients. Define $(w, \tilde{w} \in \Omega)$

$$\begin{aligned} Q(w; \tilde{w}|\theta) &\equiv \sum_{i=1}^{n} \frac{\partial g(\tilde{w}, \theta)}{\partial w_i} \frac{\tilde{w}_i}{g(\tilde{w}, \theta)} \log w_i \\ &= \sum_{i=1}^{n} tr(M^{-1}(\tilde{w}, \theta) A_i(\theta)) \tilde{w}_i \log w_i. \end{aligned}$$

By Lemma 2, we have

$$Q(w; \tilde{w}|\theta) - Q(\tilde{w}; \tilde{w}|\theta) \le \log g(w, \theta) - \log g(\tilde{w}, \theta).$$

Integration yields

$$\begin{aligned}\sum_{i=1}^{n} d_i(\tilde{w})\tilde{w}_i \log\frac{w_i}{\tilde{w}_i} &= \int \left[Q(w;\tilde{w}|\theta) - Q(\tilde{w};\tilde{w}|\theta)\right] \mathrm{d}\pi(\theta) \\ &\leq \int \left[\log g(w,\theta) - \log g(\tilde{w},\theta)\right] \mathrm{d}\pi(\theta) \\ &= \phi(w) - \phi(\tilde{w}).\end{aligned}$$

That is, the function

$$Q(w;\tilde{w}) = \sum_{i=1}^{n} d_i(\tilde{w})\tilde{w}_i \log\frac{w_i}{\tilde{w}_i} + \phi(\tilde{w})$$

satisfies $Q(w;\tilde{w}) \leq \phi(w)$ and $Q(w;w) = \phi(w)$. This forms the basis of minorization-maximization.

Suppose $w^{(t)}, w^{(t+1)} \in \Omega$ are related by (2). Applying Lemma 3 with $q_i = d_i\left(w^{(t)}\right) w_i^{(t)}/m$ (note that $\sum_{i=1}^{n} q_i = 1$), we get

$$Q\left(w^{(t+1)};w^{(t)}\right) \geq Q\left(w^{(t)};w^{(t)}\right), \tag{8}$$

as long as (3) holds. Thus $\phi\left(w^{(t+1)}\right) \geq Q\left(w^{(t+1)};w^{(t)}\right) \geq Q\left(w^{(t)};w^{(t)}\right) = \phi\left(w^{(t)}\right)$, and monotonicity is proved. Lemma 3 implies that equality holds in (8) only when $w^{(t+1)} = w^{(t)}$. Hence the monotonicity is strict. □

Remark 1. Theorem 1 assumes that $w^{(t)}, w^{(t+1)} \in \Omega$, i.e., they have all positive coordinates. This assumption can be relaxed. Inspection of the above proof shows that strict monotonicity holds as long as $w^{(t)} \in \bar{\Omega}$ and $\phi\left(w^{(t)}\right)$ is finite.

Remark 2. The arguments of Yu (2010a), based on two layers of auxiliary variables, can be extended to prove the monotonicity of (2) assuming $\alpha^{(t)} \equiv 0$. This is however weaker than Theorem 1 in the present form.

Acknowledgments

This work is partly supported by a CORCL special research grant from the University of California, Irvine. The author would like to thank Don Rubin, Xiao-Li Meng, and David van Dyk for introducing him to the field of statistical computing.

References

[1] K. Chaloner and K. Larntz, Optimal Bayesian design applied to logistic regression experiments, J. Statist. Plann. Inference 21 (1989) pp. 191-208.

[2] K. Chaloner and I. Verdinelli, Bayesian experimental design: a review, Statist. Sci. 10 (1995) pp. 273-304.

[3] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm (with discussion), J. Roy. Statist. Soc. B 39 (1977) pp. 1–38.

[4] H. Dette, A. Pepelyshev and A. Zhigljavsky, Improving updating rules in multiplicative algorithms for computing D-optimal designs, Computational Statistics & Data Analysis 53 (2008) pp. 312–320.

[5] V.V. Fedorov, Theory of Optimal Experiments, Academic Press, New York (1972).

[6] R.A. Horn and C.R. Johnson, Matrix Analysis, Cambridge University Press (1990).

[7] D.R. Hunter and K. Lange, A tutorial on MM algorithms, The American Statistician 58 (2004) pp. 30–37.

[8] J. Kiefer, General equivalence theory for optimum designs (approximate theory), Ann. Statist. 2 (1974) pp. 849–879.

[9] J. Kiefer and J. Wolfowitz, The equivalence of two extremum problems, Canad. J. Math. 12 (1960) pp. 363–366.

[10] K. Lange, D.R. Hunter and I. Yang, Optimization transfer using surrogate objective functions (with discussion), Journal of Computational and Graphical Statistics 9 (2000) pp. 1–59.

[11] D.V. Lindley, On a measure of information provided by an experiment, The Annals of Mathematical Statistics 27 (1956) pp. 986-1005.

[12] A. Pázman, Foundations of Optimum Experimental Design, Reidel, Dordrecht (1986).

[13] F. Pukelsheim, Optimal Design of Experiments, John Wiley & Sons Inc, New York (1993).

[14] S.D. Silvey, Optimal Design, Chapman & Hall, London (1980).

[15] S.D. Silvey, D.M. Titterington and B. Torsney, An algorithm for optimal designs on a finite design space, Commun. Stat. Theory Methods 14 (1978) pp. 1379-1389.

[16] D.M. Titterington, Algorithms for computing D-optimal design on finite design spaces. In Proc. of the 1976 Conf. on Information Science and Systems, John Hopkins University, 3 (1976) pp. 213-216.

[17] D.M. Titterington, Estimation of correlation coefficients by ellipsoidal trimming, Appl. Stat. 27 (1978) pp. 227-234.

[18] B. Torsney and S. Mandal, Two classes of multiplicative algorithms for constructing optimizing distributions, Computational Statistics & Data Analysis 51 (2006) pp. 1591–1601.

[19] P. Whittle, Some general points in the theory of optimal experimental design. J. R. Statist. Soc. B 35 (1973) 123–130.

[20] D. Young, *Iterative Solutions of Large Linear Systems*. New York: Academic Press (1971).

[21] Y. Yu, Monotonic convergence of a general algorithm for computing optimal designs, Annals of Statistics 38 (2010a) pp. 1593–1606. arXiv:0905.2646v3

[22] Y. Yu, Strict monotonicity and convergence rate of Titterington's algorithm for computing D-optimal designs, Computational Statistics & Data Analysis 54 (2010b) pp. 1419–1425.

[23] Y. Yu, D-optimal designs via a cocktail algorithm, Technical Report (2010c) arXiv:0911.0108

[24] Y. Yu, Monotonically overrelaxed EM algorithms, Technical Report (2010d), Department of Statistics, University of California, Irvine.

[25] Y. Yu, Improved EM for mixture proportions with applications to nonparametric ML estimation for censored data, Technical Report (2010e) arXiv:1002.3640