Lecture #13 Notes Summary

Classical algorithms for the computation of optimal designs

Besides the semidefinite programming approach for the computation of approximate optimal designs, other algorithms exist (and have been developped earlier !). The objective of this lecture is to review some features of these algorithms, for the computation of exact and approximate optimal designs.

We will focus on the case of $D-$optimality, but these algorithms can often be adapted to the case of $A-$optimality (and sometimes $E-$optimality but this is harder, because λ_{min} is not differentiable, only sub-differentiable).

Fedorov-Wynn algorithm

Recall the Kiefer-Wolfowitz equivalence theorem: an approximate design is $D-$optimal iff for all $\boldsymbol{x}$ in $\mathcal{X}$, $\boldsymbol{a}(\boldsymbol{x})^T M(\xi)^- \boldsymbol{a}(\boldsymbol{x}) \leq m$.

The Fedorov-Wynn (FW) algorithm tries to find a design ξ satisfying this property. It starts with an arbitrary feasible design $\xi_0 \in \Xi := \Xi(\boldsymbol{I_m})$, for example a design with weights $w_i = \frac{1}{m}$ on points $\boldsymbol{x_i}$ such that the regression vectors $\boldsymbol{a}(\boldsymbol{x_1}), \ldots, \boldsymbol{a}(\boldsymbol{x_m})$ are independent. At the k^{th} step of the algorithm, the current design is ξ_k, and the goal is to identify a point $\boldsymbol{x}^{(k)} \in \mathcal{X}$ that solves

$$\max_{\boldsymbol{x} \in \mathcal{X}} \ \boldsymbol{a}(\boldsymbol{x})^T M(\xi_k)^- \boldsymbol{a}(\boldsymbol{x}) \tag{1}$$

Note that Problem (1) can be complicated to solve, but in practice the set $\mathcal{X}$ is discretized as $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_s}\}$, so that Problem 1 reduces to a maximum over a finite set. Then, the design is updated by moving in the direction of $\boldsymbol{x}^{(k)}$:

$$\xi_{k+1} = (1 - \alpha_k)\xi_k + \alpha_k \xi(\boldsymbol{x}^{(k)}), \tag{2}$$

where $\xi(\boldsymbol{x})$ denotes the design with 100% of the weight concentrated at $\boldsymbol{x}$. For particular step sizes α_k, it is known that this process converges to the $D-$optimal design. We are going to prove this for the Fedorov rule

$$\alpha_k := \frac{d_k - m}{m(d_k - 1)}, \tag{3}$$

where $d_k := \boldsymbol{x}^{(k)} M(\xi_k)^- \boldsymbol{x}^{(k)}$. In fact, Wynn proved that the above algorithm converges to the $D-$optimal design for all sequences of step lengths α_k satisfying $\lim_{k \to \infty} \alpha_k = 0$ and $\sum_k^\infty \alpha_k = +\infty$.

The proof relies on the following lemma.

Lemma 1 (Matrix Determinant Lemma). *Let A be an invertible matrix of size $n \times n$, and let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two vectors of size $\boldsymbol{n}$. Then,*

$$\det(A + \boldsymbol{u}\boldsymbol{v}^T) = \det A \, (1 + \boldsymbol{v}^T A^{-1} \boldsymbol{u}).$$

Proof. The proof in the case $A = \boldsymbol{I}$ follows from

$$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{v}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{u} \\ \boldsymbol{0}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{v}^T & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{u} \\ \boldsymbol{0}^T & 1 + \boldsymbol{v}^T \boldsymbol{u} \end{pmatrix}.$$

Then, for an invertible matrix A we have

$$\det(A + \boldsymbol{u}\boldsymbol{v}^T) = \det A \, \det(\boldsymbol{I} + A^{-1}\boldsymbol{u}\boldsymbol{v}^T) = \det A \, (1 + \boldsymbol{v}^T A^{-1} \boldsymbol{u}).$$

□

Proposition 2. *Let $\xi_k \in \Xi$. The Fedorov rule* (3) *defines optimal step sizes, in the sense that α_k maximizes* $\det M(\xi_{k+1})$*.*

Proof. We can write $M(\xi_{k+1})$ as

$$M(\xi_{k+1}) = (1-\alpha_k)\left(M(\xi_k) + \frac{\alpha_k}{1-\alpha_k}\boldsymbol{a}(\boldsymbol{x}_k)\boldsymbol{a}(\boldsymbol{x}_k)^T\right).$$

This form makes it possible to use the matrix determinant lemma:

$$\det M(\xi_{k+1}) = (1-\alpha_k)^m \left(1 + \frac{\alpha_k}{1-\alpha_k} d_k\right) \det M(\xi_k),$$

where d_k has been set to $\boldsymbol{a}(\boldsymbol{x}^{(k)})M(\xi_k)^{-}\boldsymbol{a}(\boldsymbol{x}^{(k)})$. We know from the KW theorem that $d_k \geq m$ (with equality iff ξ_k is $D-$optimal). So the step size α_k defined in (3) lies in the interval $[0,1]$, and it is easy to check that α_k maximizes $\det M(\xi_{k+1})M(\xi_k)^{-1}$ by differentiating the expression $(1-\alpha_k)^m\left(1+\frac{\alpha_k}{1-\alpha_k}d_k\right)$ with respect to α_k. □

We can now prove the following theorem:

Theorem 3. *Let $\xi_0, \xi_1, \dots, \xi_k, \dots$ be a sequence of designs generated by the FW algorithm, with $\xi_0 \in \Xi$ and step sizes α_k following rule* (3)*. Then, $\xi_k \in \Xi$ for all k, the sequence of $D-$criterions is nondecreasing:*

$$\Phi_D\big(M(\xi_0)\big) \leq \Phi_D\big(M(\xi_1)\big) \leq \Phi_D\big(M(\xi_2)\big) \leq \dots,$$

and (ξ_k) converges to a $D-$optimal design.

Proof. The new design ξ_{k+1} is the barycenter of 2 designs so it is a design. Its feasibility ($\xi_{k+1} \in \Xi$) is a consequence from the fact that the sequence $\Big(\det M(\xi_k)\Big)$ is nondecreasing, which follows from Proposition 2.

So, the only thing to prove is that (ξ_k) converges to an optimal design. The sequence $\Big(\det M(\xi_k)\Big)$ is nondecreasing and is bounded from above by $D^* = \det M(\xi^*)$, where ξ^* is a $D-$optimal design, so it has a limit. Assume *ad absurbium* that $\lim_{k\to\infty} \det M(\xi_k) = D < D^*$. This implies $\liminf_{k\to\infty} d_k = m + \delta$ for some $\delta > 0$ (otherwise, there would exist a subsequence of designs such that $d_k \to m$ and $\det M(\xi_k) \not\to D^*$, which would contradict the KW theorem). Now, substituting the value of α_k in the expression of $\det M(\xi_{k+1})M(\xi_{k+1})^{-1}$, we find (after some simplifications):

$$\det M(\xi_{k+1})M(\xi_{k+1})^{-1} = (1-\alpha_k)^m\left(1+\frac{\alpha_k}{1-\alpha_k}d_k\right) = \left(\frac{d}{m}\right)^m \left(\frac{m-1}{d-1}\right)^{m-1}.$$

The log of this expression can be written as $f(d) - f(m)$, where $f : x \mapsto m\log x - (m-1)\log(x-1)$ is an increasing function of x for $x \geq m$. This shows

$$\log\det M(\xi_{k+1}) - \log\det M(\xi_{k+1}) \geq f(m+\delta) - f(m) > 0,$$

and $\det M(\xi_k) \to \infty$. Contradiction. □

Remark 4. In practice, this algorithm has a rather slow rate of convergence. But several techniques can be used to accelerate it, without loosing the theoretical convergence result. For example, one can *remove weight* from the support point $\boldsymbol{x_i}$ of ξ_k minimizing $\boldsymbol{a}(\boldsymbol{x_i})^T M(\xi_k)^- \boldsymbol{a}(\boldsymbol{x_i})$, or *merge* close support points. There is also an important result which allows to remove points that cannot support the optimal design, which yields a very important speed-up. We'll prove this result in exercises:

Theorem 5 (Harman & Pronzato)**.** *Let $\xi \in \Xi$, and define $\epsilon := \max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{a}(\boldsymbol{x})^T M(\xi)^- \boldsymbol{a}(\boldsymbol{x}) - m$. If a point $\boldsymbol{x} \in \mathcal{X}$ is such that*

$$\boldsymbol{a}(\boldsymbol{x})^T M(\xi)^- \boldsymbol{a}(\boldsymbol{x}) < m \left(1 + \frac{\epsilon}{2} - \frac{\sqrt{\epsilon(4+\epsilon-4/m)}}{2}\right),$$

then $\boldsymbol{x}$ cannot belong to the support of any $D-$optimal design.

Heuristics for the construction of exact optimal designs

Many heuristics have been proposed to construct exact optimal designs. Besides the rounding heuristics based on the computation of an approximate optimal design (which are known to be very efficient when the total number of trials N is large), many are based on neighbourhood searches, such as the KL-exchange procedure of Aktinson & Donev. We do not enter the details of these procedures (basically, any standard heuristic that explore the neighbourhood of some solutions, such as simulated annealing, can be used), but we show below that it is possible to evaluate efficienty the increase of Φ_D when the design point i is replaced by the design point j.

The basic idea is to use the *Sherman-Morrison* formula, which explain how to compute rank-one updates of a matrix inverse. Note that this formula can also be used to make an efficient implementation of the FW algorithm for approximate designs.

Lemma 6 (Sherman-Morrison formula)**.** *Let A be an invertible matrix of size $n \times n$, and let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two vectors of size $\boldsymbol{n}$. Then,*

$$(A + \boldsymbol{u}\boldsymbol{v}^T)^{-1} = A^{-1} - \frac{(A^{-1}\boldsymbol{u})(A^{-1}\boldsymbol{v})^T}{1 + \boldsymbol{v}^T A^{-1}\boldsymbol{u}}.$$

Proof. let $X = A + \boldsymbol{u}\boldsymbol{v}^T$, $Y = A^{-1} - \frac{(A^{-1}\boldsymbol{u})(A^{-1}\boldsymbol{v})^T}{1+\boldsymbol{v}^T A^{-1}\boldsymbol{u}}$, and simply check that $XY = YX = \boldsymbol{I}$. □

Proposition 7. *Let $\xi \in \Xi_N$ be a feasible exact design of size N, and let ξ' denote the design obtained by replacing one observation at design point $\boldsymbol{x_i}$ by one observation at $\boldsymbol{x_j}$. Let $M = M(\xi)$, $M = M(\xi')$, $\boldsymbol{a}_i = \boldsymbol{a}(\boldsymbol{x_i})$, $\boldsymbol{a}_j = \boldsymbol{a}(\boldsymbol{x_j})$, $d_i = \boldsymbol{a}_i^T M^- \boldsymbol{a}_i$, $d_j = \boldsymbol{a}_j^T M^- \boldsymbol{a}_j$, and $d_{ij} = \boldsymbol{a}_i^T M^- \boldsymbol{a}_j$. Then,*

$$\det M' = \det M \quad \Big((1+d_j)(1-d_i) + d_{ij}^2\Big).$$

Proof. Let $M^+ = M + \boldsymbol{a}_j\boldsymbol{a}_j^T$, so that $M' = M^+ - \boldsymbol{a}_i\boldsymbol{a}_i^T$. From the matrix determinant lemma, we have $\det M^+ = \det M\ (1+d_j)$ and $\det M' = \det M^+\ (1 - \boldsymbol{a}_i^T (M^+)^{-1}\boldsymbol{a}_i^T)$. Now we use the Sherman-Morrison formula:

$$(M^+)^{-1} = M^{-1} - \frac{(M^{-1}\boldsymbol{a_j})(M^{-1}\boldsymbol{a_j})^T}{1+d_j},$$

so that $\boldsymbol{a}_i^T (M^+)^{-1}\boldsymbol{a}_i^T = d_i - \frac{d_{ij}^2}{1+d_j}$. We can now obtain the formula of the proposition:

$$\det M' = \det M^+\ (1 - \boldsymbol{a}_i^T (M^+)^{-1}\boldsymbol{a}_i^T) = \det M\ (1+d_j)(1 - d_i + \frac{d_{ij}^2}{1+d_j}) = \det M \quad \Big((1+d_j)(1-d_i) + d_{ij}^2\Big).$$

□

Exercises

Proof of Theorem 5. Let $\xi \in \Xi$ and ξ^* be a $D-$optimal design. Define $M = M(\xi), M^* = M(\xi^*)$, and $H = M^{-1/2} M^* M^{-1/2}$. Let $\boldsymbol{x^*}$ be a support point of ξ^*, and define $\boldsymbol{y} = H^{-1/2} M^{-1/2} \boldsymbol{a}(\boldsymbol{x^*})$.

1. What is the value of $\boldsymbol{a}(\boldsymbol{x^*})^T \, (M^*)^{-1} \, \boldsymbol{a}(\boldsymbol{x^*})$?
2. Compute $\boldsymbol{y}^T \boldsymbol{y}$, $\boldsymbol{y}^T H \boldsymbol{y}$, and deduce that

$$\boldsymbol{a}(\boldsymbol{x^*})^T M^{-1} \boldsymbol{a}(\boldsymbol{x^*}) \geq \lambda_1 m,$$

   where λ_1 is the smallest eigenvalue of H.

   To prove the theorem, we must find a lower bound of λ_1, which does not depend on M^*.

3. Set $\epsilon := \max_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{a}(\boldsymbol{x})^T M(\xi)^- \boldsymbol{a}(\boldsymbol{x}) - m$. Show that $\operatorname{trace} H^{-1} \leq m$ and $\operatorname{trace} H \leq m + \epsilon$.
4. Deduce that λ_1 is bounded from below by the optimal value of

$$\begin{aligned} \min \quad & \lambda_1 \\ s.t. \quad & \sum_{i=1}^{m} \lambda_i \leq m + \epsilon \\ & \sum_{i=1}^{m} \lambda_i^{-1} \leq m \end{aligned} \tag{4}$$

5. (harder) Use an argument involving the Lagrangian of Problem (4) to show that at optimality, we must have $\lambda_2 = \lambda_3 = \ldots = \lambda_m$.
6. You can now compute the optimal value of Problem (4). Show that this reduces to solving a system of 2 equations for the variables λ_1 and λ_2. You can verify that the smallest solution λ_1 of this system is

$$\lambda_1 = \left(1 + \frac{\epsilon}{2} - \frac{\sqrt{\epsilon(4 + \epsilon - 4/m)}}{2} \right).$$

   The theorem follows.