A Maximin Φ_p-Efficient Design for Multivariate Generalized Linear Models

Yiou Li, Lulu Kang and Xinwei Deng

DePaul University, Illinois Institute of Technology, and Virginia Tech

Supplementary Material

S1 Lemmas and Proofs of Theorems

Lemma 1. $\mathrm{EA}(\cdot;\mathcal{M}')$ defined in (2.9) is a convex function of design ξ.

Proof. Since $\Phi_p(\cdot;\mathcal{M}')$ is convex with respect to ξ (Yang and Stufken, 2012) and $\exp(\cdot)$ is a convex and strictly increasing function, the composite function $\xi \mapsto \exp\left(\frac{\Phi_p^j(\xi)}{\Phi_p^{\mathrm{opt}_j}}\right)$ is a convex function of ξ. As a result, $\mathrm{EA}(\cdot;\mathcal{M}')$ is a convex function of ξ. $\square$

Lemma 2. The directional derivative of $\mathrm{EA}(\xi;\mathcal{M}')$ in the direction from ξ to ξ' is

$$\text{when } p=0, \quad \phi(\xi',\xi) = \sum_{j=1}^{m} \tilde{\Phi}_0^j(\xi)\left[q - \mathrm{tr}\left(\mathsf{F}_j(\xi)^{-1}\mathsf{G}_j(\xi,\xi')\right)\right],$$

$$\text{when } p>0, \quad \phi(\xi',\xi) = \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi)\left[\Phi_p^j(\xi) - q^{-1/p}\left(\mathrm{tr}\left(\mathsf{F}_j(\xi)\right)^p\right)^{1/p-1}\mathrm{tr}\left(\left(\mathsf{F}_j(\xi)\right)^{p-1}\mathsf{G}_j(\xi,\xi')\right)\right],$$

where

$$\tilde{\Phi}_p^j(\xi) = \left[\Phi_p^{\mathrm{opt}_j}\right]^{-1} \exp\left(\frac{\Phi_p^j(\xi)}{\Phi_p^{\mathrm{opt}_j}}\right), \quad \mathsf{B}_j = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^\top}\right|_{\boldsymbol{\beta}=\boldsymbol{\beta}_j},$$

$$\mathsf{F}_j(\xi) = \mathsf{B}_j \mathsf{I}_j(\xi)^{-1}\mathsf{B}_j^\top, \quad \mathsf{G}_j(\xi,\xi') = \mathsf{B}_j \mathsf{I}_j(\xi)^{-1}\mathsf{I}_j(\xi')\mathsf{I}_j(\xi)^{-1}\mathsf{B}_j^\top.$$

Proof. Given $\tilde{\xi} = (1-\alpha)\xi + \alpha\xi'$, we have $\mathsf{I}_j(\tilde{\xi}) = (1-\alpha)\mathsf{I}_j(\xi) + \alpha\mathsf{I}_j(\xi')$. For any invertible matrix S, whose elements are functions of α, the derivative of S^{-1} is $\frac{\partial \mathsf{S}^{-1}}{\partial\alpha} = -\mathsf{S}^{-1}\frac{\partial \mathsf{S}}{\partial\alpha}\mathsf{S}^{-1}$. So, the derivative of $\mathsf{I}_j(\tilde{\xi})^{-1}$ with respect to α can be expressed as,

$$\frac{\partial\left[\mathsf{I}_j(\tilde{\xi})^{-1}\right]}{\partial\alpha} = -\mathsf{I}_j(\tilde{\xi})^{-1}[\mathsf{I}_j(\xi') - \mathsf{I}_j(\xi)]\mathsf{I}_j(\tilde{\xi})^{-1}. \tag{S1.1}$$

Thus, for $p = 0$,

$$\begin{aligned}\frac{\partial\Phi_0^j(\tilde{\xi})}{\partial\alpha} &= \frac{\partial \log\left|\mathsf{F}_j(\tilde{\xi})\right|}{\partial\alpha} = \mathrm{tr}\left[\mathsf{F}_j(\tilde{\xi})^{-1}\mathsf{B}_j\frac{\partial\left[\mathsf{I}_j(\tilde{\xi})^{-1}\right]}{\partial\alpha}\mathsf{B}_j^\top\right] \\ &= -\,\mathrm{tr}\left[\mathsf{F}_j(\tilde{\xi})^{-1}\mathsf{B}_j\mathsf{I}_j(\tilde{\xi})^{-1}[\mathsf{I}_j(\xi') - \mathsf{I}_j(\xi)]\mathsf{I}_j(\tilde{\xi})^{-1}\mathsf{B}_j^\top\right];\end{aligned} \tag{S1.2}$$

for $p > 0$,

$$\begin{aligned}\frac{\partial\Phi_p^j(\tilde{\xi})}{\partial\alpha} &= \frac{\partial\left[\left(q^{-1}\,\mathrm{tr}\left(\mathsf{F}_j(\tilde{\xi})\right)^p\right)^{1/p}\right]}{\partial\alpha} \\ &= -q^{-1/p}\left(\mathrm{tr}\left(\mathsf{F}_j(\tilde{\xi})\right)^p\right)^{1/p-1}\mathrm{tr}\left[\left(\mathsf{F}_j(\tilde{\xi})\right)^{p-1}\mathsf{B}_j\mathsf{I}_j(\tilde{\xi})^{-1}[\mathsf{I}_j(\xi') - \mathsf{I}_j(\xi)]\mathsf{I}_j(\tilde{\xi})^{-1}\mathsf{B}_j^\top\right].\end{aligned} \tag{S1.3}$$

Based on (S1.2) and (S1.3), the directional derivative of $\mathrm{EA}(\xi, \mathcal{M}')$ is

$$\phi(\xi', \xi) = \left.\frac{\partial \left[\sum_{j=1}^{m} \exp\left(\frac{\Phi_p^j(\tilde{\xi})}{\Phi_p^{\mathrm{opt}_j}}\right)\right]}{\partial \alpha}\right|_{\alpha=0} = \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi) \left.\frac{\partial \Phi_p^j(\tilde{\xi})}{\partial \alpha}\right|_{\alpha=0}$$

$$= \begin{cases} \sum_{j=1}^{m} \tilde{\Phi}_0^j(\xi) \left[q - \mathrm{tr}\left(\mathsf{F}_j(\xi)^{-1}\mathsf{G}_j(\xi, \xi')\right)\right], & p = 0; \\ \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi) \left[\Phi_p^j(\xi) - q^{-1/p}\left(\mathrm{tr}\left(\mathsf{F}_j(\xi)\right)^p\right)^{1/p-1} \mathrm{tr}\left(\left(\mathsf{F}_j(\xi)\right)^{p-1}\mathsf{G}_j(\xi, \xi')\right)\right], & p > 0. \end{cases}$$

$\square$

Lemma 3. The directional derivative of $\mathrm{EA}(\xi; \mathcal{M}')$ in the direction of a single point $\boldsymbol{x}$ is given as,

$$\phi(\boldsymbol{x}, \xi) = \begin{cases} \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi) \left[q - w_j(\boldsymbol{x})\boldsymbol{g}_j^\top(\boldsymbol{x})\mathsf{M}_j(\xi)\boldsymbol{g}_j(\boldsymbol{x})\right], & p = 0; \\ \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi) \left[\Phi_p^j(\xi) - q^{-1/p} w_j(\boldsymbol{x})\left(\mathrm{tr}\left(\mathsf{F}_j(\xi)\right)^p\right)^{1/p-1} \boldsymbol{g}_j^\top(\boldsymbol{x})\mathsf{M}_j(\xi)\boldsymbol{g}_j(\boldsymbol{x})\right], & p > 0, \end{cases}$$

where $\mathsf{M}_j(\xi) = \mathsf{I}_j(\xi)^{-1}\mathsf{B}_j^\top \mathsf{F}_j(\xi)^{p-1}\mathsf{B}_j \mathsf{I}_j(\xi)^{-1}$.

Particularly, the directional derivatives of D-, A- and the prediction-oriented criterion EI-optimality defined in (Li and Deng, 2020) are:

$$\phi(\boldsymbol{x}, \xi) = \begin{cases} \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi) \left[l - w_j(\boldsymbol{x})\boldsymbol{g}_j^\top(\boldsymbol{x})\mathsf{I}_j^{-1}(\xi)\boldsymbol{g}_j(\boldsymbol{x})\right], & \text{D-optimality;} \\ \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi) \left[\mathrm{tr}(\mathsf{I}_j^{-1}(\xi)) - w_j(\boldsymbol{x})\boldsymbol{g}_j^\top(\boldsymbol{x})\mathsf{I}_j^{-2}(\xi)\boldsymbol{g}_j(\boldsymbol{x})\right], & \text{A-optimality;} \\ \sum_{j=1}^{m} \tilde{\Phi}_p^j(\xi) \left[\mathrm{tr}(\mathsf{I}_j^{-1}(\xi)\mathsf{A}_j) - w_j(\boldsymbol{x})\boldsymbol{g}_j^\top(\boldsymbol{x})\mathsf{I}_j^{-1}(\xi)\mathsf{A}_j\mathsf{I}_j^{-1}(\xi)\boldsymbol{g}_j(\boldsymbol{x})\right], & \text{EI-optimality,} \end{cases}$$

where $\mathsf{A}_j = \int_\Omega \boldsymbol{g}_j(\boldsymbol{x})\boldsymbol{g}_j^\top(\boldsymbol{x})\left[\frac{\mathrm{d}h_j^{-1}}{\mathrm{d}\eta_j}\right]^2 \mathrm{d}F_{\mathrm{IMSE}}(\boldsymbol{x})$ for EI-optimality is pre-determined and it does not depend on the design ξ. The cdf F_{IMSE} is the user-specified distribution for the EI-optimality.

Lemma 4. For a design $\xi^{\boldsymbol{\lambda}}$ with fixed design points $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$, $\text{EA}(\cdot; \mathcal{M}')$ in (2.11) is a convex function with respect to the weight vector $\boldsymbol{\lambda}$.

Proof. The proof is similar to that of Lemma 1. □

Proof of Theorem 1

Proof. i. (1) → (2): As $\text{EA}(\cdot; \mathcal{M}')$ is a convex function in ξ proved in Lemma 1, the directional derivative $\phi(\boldsymbol{x}, \xi^{\text{Mm}}_{\mathcal{M}'}) \geq 0$ holds for any $\boldsymbol{x} \in \Omega$, and the inequality becomes equality if $\boldsymbol{x}$ is a support point of the design $\xi^{\text{Mm}}_{\mathcal{M}'}$.

ii. (2) → (1): If $\phi(\boldsymbol{x}, \xi^{\text{Mm}}_{\mathcal{M}'}) \geq 0$ holds for any $\boldsymbol{x} \in \Omega$, then $\xi^{\text{Mm}}_{\mathcal{M}'}$ minimizes $\text{EA}(\xi; \mathcal{M}')$ as $\text{EA}(\cdot; \mathcal{M}')$ is a convex function in ξ.

□

Proof of Theorem 2

We first establish the proof of the following lemma.

Lemma 5. For any design ξ and the Mm-Φ_p design $\xi^{\text{Mm}}_{\mathcal{M}'}$ that minimizes $\text{EA}(\xi; \mathcal{M}')$ or equivalently minimizes $\text{LEA}(\xi; \mathcal{M}')$, the following inequality holds:

$$\min_{\boldsymbol{x}\in\Omega}\phi(\boldsymbol{x},\xi)\leq\phi(\xi_{\mathcal{M}'}^{\mathrm{Mm}},\xi)\leq\mathrm{EA}(\xi_{\mathcal{M}'}^{\mathrm{Mm}};\mathcal{M}')-\mathrm{EA}(\xi;\mathcal{M}')\leq 0,$$

where $\phi(\boldsymbol{x},\xi)$ and $\phi(\xi_{\mathcal{M}'}^{\mathrm{Mm}},\xi)$ are the directional derivatives defined in (2.10).

Proof. The lemma is proved for the $p>0$ case, and the case $p=0$ could be proved similarly. The directional derivative of $\mathrm{EA}(\xi;\mathcal{M}')$ in the direction of the point $\boldsymbol{x}^*=\underset{\boldsymbol{x}\in\Omega}{\operatorname{argmin}}\,\phi(\boldsymbol{x},\xi)$ is:

$$\begin{aligned}
&\min_{\boldsymbol{x}\in\Omega}\phi(\boldsymbol{x},\xi)=\phi(\boldsymbol{x}^*,\xi)\\
=\;&\sum_{j=1}^{m}\tilde{\Phi}_j(\xi)\left[\Phi_p^j(\xi)-q^{-1/p}\left(\operatorname{tr}\left(\mathsf{F}_j(\xi)\right)^p\right)^{1/p-1}\operatorname{tr}\left(\left(\mathsf{F}_j(\xi)\right)^{p-1}\mathsf{B}_j\mathsf{I}_j(\xi^{(r)})^{-1}\mathsf{I}_j(\boldsymbol{x}_r^*)\mathsf{I}_j(\xi)^{-1}\mathsf{B}_j^\top\right)\right]\\
\leq\;&\sum_{j=1}^{m}\tilde{\Phi}_j(\xi)\left[\Phi_p^j(\xi)-q^{-1/p}\left(\operatorname{tr}\left(\mathsf{F}_j(\xi)\right)^p\right)^{1/p-1}\operatorname{tr}\left(\left(\mathsf{F}_j(\xi)\right)^{p-1}\mathsf{B}_j\mathsf{I}_j(\xi)^{-1}\mathsf{I}_j(\boldsymbol{z})\mathsf{I}_j(\xi)^{-1}\mathsf{B}_j^\top\right)\right]
\end{aligned}\tag{S1.4}$$

for any $\boldsymbol{x}\in\Omega$, where $\mathsf{I}_j(\boldsymbol{x})$ denotes the information matrix of the design with a unit mass on single point $\boldsymbol{x}$.

Denote the Mm-Φ_p design $\xi_{\mathcal{M}'}^{\mathrm{Mm}}=\left\{\begin{array}{ccc}\boldsymbol{x}_1, & ..., & \boldsymbol{x}_n\\ \lambda_1^*, & ..., & \lambda_n^*\end{array}\right\}$. With (S1.4), we have

$$
\begin{aligned}
& \phi(\boldsymbol{x}^*, \xi) \\
\leq \; & \sum_{i=1}^{n} \lambda_i^* \sum_{j=1}^{m} \tilde{\Phi}_j(\xi) \left[\Phi_p^j(\xi) - q^{-1/p} \left(\operatorname{tr} \left(\mathsf{F}_j(\xi) \right)^p \right)^{1/p-1} \operatorname{tr} \left(\left(\mathsf{F}_j(\xi) \right)^{p-1} \mathsf{B}_j \mathsf{I}_j(\xi)^{-1} \mathsf{I}_j(\boldsymbol{x}_i) \mathsf{I}_j(\xi)^{-1} \mathsf{B}_j^\top \right) \right] \\
= \; & \sum_{j=1}^{m} \tilde{\Phi}_j(\xi) \left[\Phi_p^j(\xi) - q^{-1/p} \left(\operatorname{tr} \left(\mathsf{F}_j(\xi) \right)^p \right)^{1/p-1} \operatorname{tr} \left(\left(\mathsf{F}_j(\xi) \right)^{p-1} \mathsf{B}_j \mathsf{I}_j(\xi)^{-1} \mathsf{I}_j(\xi_{\mathcal{M}'}^{\mathrm{Mm}}) \mathsf{I}_j(\xi)^{-1} \mathsf{B}_j^\top \right) \right] \\
= \; & \phi(\xi_{\mathcal{M}'}^{\mathrm{Mm}}, \xi) \qquad \text{(S1.5)}
\end{aligned}
$$

Furthermore, with the definition of directional derivative in the direction of the Mm-Φ_p design $\xi_{\mathcal{M}'}^{\mathrm{Mm}}$ and convexity of $\mathrm{EA}(\cdot;\mathcal{M}')$, we have

$$
\begin{aligned}
\phi(\xi_{\mathcal{M}'}^{\mathrm{Mm}}, \xi) &= \lim_{\alpha \to 0} \frac{\mathrm{EA}((1-\alpha)\xi + \alpha \xi_{\mathcal{M}'}^{\mathrm{Mm}}; \mathcal{M}') - \mathrm{EA}(\xi; \mathcal{M}')}{\alpha} \\
&\leq \lim_{\alpha \to 0} \frac{(1-\alpha)\,\mathrm{EA}(\xi; \mathcal{M}') + \alpha\,\mathrm{EA}(\xi_{\mathcal{M}'}^{\mathrm{Mm}}; \mathcal{M}') - \mathrm{EA}(\xi; \mathcal{M}')}{\alpha} \\
&= \mathrm{EA}(\xi_{\mathcal{M}'}^{\mathrm{Mm}}; \mathcal{M}') - \mathrm{EA}(\xi; \mathcal{M}') \qquad \text{(S1.6)}
\end{aligned}
$$

Combining (S1.5) and (S1.6), we complete the proof that

$$
\min_{\boldsymbol{x} \in \Omega} \phi(\boldsymbol{x}, \xi) \leq \phi(\xi_{\mathcal{M}'}^{\mathrm{Mm}}, \xi) \leq \mathrm{EA}(\xi_{\mathcal{M}'}^{\mathrm{Mm}}; \mathcal{M}') - \mathrm{EA}(\xi; \mathcal{M}') \leq 0.
$$

□

Proof. When $1 + 2\frac{\min_{\boldsymbol{x}\in\Omega} \phi(\boldsymbol{x},\xi)}{\mathrm{EA}(\xi;\mathcal{M}')} < 0$, $\mathrm{Eff}_{\mathrm{LEA}}(\xi, \xi_{\mathcal{M}'}^{\mathrm{Mm}}; \mathcal{M}') \geq 1 + 2\frac{\min_{\boldsymbol{x}\in\Omega} \phi(\boldsymbol{x},\xi)}{\mathrm{EA}(\xi;\mathcal{M}')}$ holds automatically.

When $1 + 2\frac{\min_{\boldsymbol{x}\in\Omega} \phi(\boldsymbol{x},\xi)}{\mathrm{EA}(\xi;\mathcal{M}')} \geq 0$, that is, $\frac{\min_{\boldsymbol{x}\in\Omega} \phi(\boldsymbol{x},\xi)}{\mathrm{EA}(\xi;\mathcal{M}')} \geq -0.5$, define $\frac{\mathrm{EA}(\xi_{\mathcal{M}'}^{\mathrm{Mm}};\mathcal{M}')}{\mathrm{EA}(\xi;\mathcal{M}')} =$

$a > 0$, then it follows immediately from Lemma 5 that

$$1 \geq a = \frac{\mathrm{EA}(\xi^{\mathrm{Mm}}_{\mathcal{M}'}, \mathcal{M}')}{\mathrm{EA}(\xi; \mathcal{M}')} \geq 1 + \frac{\min_{\boldsymbol{x} \in \Omega} \phi(\boldsymbol{x}, \xi)}{\mathrm{EA}(\xi; \mathcal{M}')} \geq 0.5. \tag{S1.7}$$

Since the function $\frac{\ln(a)}{\ln(\mathrm{EA}(\xi;\mathcal{M}'))} + 1 - a$ is an increasing function of $\mathrm{EA}(\xi; \mathcal{M}')$, $\mathrm{EA}(\xi; \mathcal{M}') \geq e$ because of its definition and $a \geq 0.5$, we have

$$\begin{aligned} & \left| \mathrm{Eff}_{\mathrm{LEA}}(\xi, \xi^{\mathrm{Mm}}_{\mathcal{M}'}; \mathcal{M}') - \frac{\mathrm{EA}(\xi^{\mathrm{Mm}}_{\mathcal{M}'}; \mathcal{M}')}{\mathrm{EA}(\xi; \mathcal{M}')} \right| = \left| \frac{\ln(a\,\mathrm{EA}(\xi; \mathcal{M}'))}{\ln(\mathrm{EA}(\xi; \mathcal{M}'))} - \frac{a\,\mathrm{EA}(\xi; \mathcal{M}')}{\mathrm{EA}(\xi; \mathcal{M}')} \right| \\ = & \left| \frac{\ln(a)}{\ln(\mathrm{EA}(\xi; \mathcal{M}'))} + 1 - a \right| \leq \max(\left|\ln(a) + 1 - a\right|, \left|1 - a\right|) \\ = & \max(-\ln(a) - 1 + a, 1 - a) = 1 - a. \end{aligned}$$

Thus, together with (S1.7), $\mathrm{Eff}_{\mathrm{LEA}}(\xi, \xi^{\mathrm{Mm}}_{\mathcal{M}'}; \mathcal{M}') \geq 2a - 1 \geq 1 + 2\frac{\min_{\boldsymbol{x} \in \Omega} \phi(\boldsymbol{x}, \xi)}{\mathrm{EA}(\xi; \mathcal{M}')}$.

□

Proof of Theorem 3

Proof. We show the proof for the scenario $p > 0$ in the Φ_p-criterion. The proof for $p = 0$ could be done similarly. The proof is established by proof of contradiction. Suppose that the Algorithm 1 does not converge to the Mm-Φ_p design ξ^*, then we have

$$\lim_{r \to \infty} \mathrm{EA}(\xi^{(r)}; \mathcal{M}') > \mathrm{EA}(\xi^{\mathrm{Mm}}_{\mathcal{M}'}; \mathcal{M}').$$

For any iteration $r+1 \geq 1$, since $\mathcal{X}^{(r)} \subset \mathcal{X}^{(r+1)}$ and the Optimal-Weight Procedure returns optimal weight vector that minimizes EA criterion, the design $\xi^{(r+1)}$ cannot be worse than the design in the previous iteration $\xi^{(r)}$, i.e.,

$$\mathrm{EA}(\xi^{(r+1)};\mathcal{M}') \leq \mathrm{EA}(\xi^{(r)};\mathcal{M}').$$

Thus, for all $r \geq 0$, there exists $a > 0$, such that,

$$\mathrm{EA}(\xi^{(r)};\mathcal{M}') > \mathrm{EA}(\xi^{\mathrm{Mm}}_{\mathcal{M}'};\mathcal{M}') + a.$$

According to Lemma 5,

$$\phi(\boldsymbol{x}_r^*,\xi^{(r)}) \leq \phi(\xi^{\mathrm{Mm}}_{\mathcal{M}'},\xi^{(r)}) \leq \mathrm{EA}(\xi^{\mathrm{Mm}}_{\mathcal{M}'};\mathcal{M}') - \mathrm{EA}(\xi^{(r)};\mathcal{M}') < -a,$$

for any $r \geq 0$. Then, the Taylor expansion of $\mathrm{EA}((1-\alpha)\xi^{(r)} + \alpha\boldsymbol{x}_r^*;\mathcal{M}')$ is upper bounded by

$$\begin{aligned}\mathrm{EA}((1-\alpha)\xi^{(r)} + \alpha\boldsymbol{x}_r^*;\mathcal{M}') &= \mathrm{EA}(\xi^{(r)};\mathcal{M}') + \phi(\boldsymbol{x}_r^*,\xi^{(r)})\alpha + \frac{u}{2}\alpha^2 \\ &< \mathrm{EA}(\xi^{(r)};\mathcal{M}') - a\alpha + \frac{u}{2}\alpha^2, \qquad \text{(S1.8)}\end{aligned}$$

where $u \geq 0$ is the second-order directional derivative of EA evaluated at a value between 0 and α.

For Algorithm 1, the criterion EA is minimized, for all $0 \leq \alpha \leq 1$ we have

$$\begin{aligned}\mathrm{EA}(\xi^{(r+1)};\mathcal{M}') &\leq \mathrm{EA}((1-\alpha)\xi^{(r)} + \alpha\boldsymbol{x}_r^*;\mathcal{M}') \\ &< \mathrm{EA}(\xi^{(r)};\mathcal{M}') - a\alpha + \frac{u}{2}\alpha^2,\end{aligned}$$

or equivalently,

$$\mathrm{EA}(\xi^{(r+1)};\mathcal{M}') - \mathrm{EA}(\xi^{(r)};\mathcal{M}') < -a\alpha + \frac{u}{2}\alpha^2 = \frac{u}{2}\left(\alpha - \frac{a}{u}\right)^2 - \frac{a^2}{2u}$$

$$< \begin{cases} -\frac{a^2}{2u} < 0, & \text{choosing } \alpha = \frac{a}{u} \text{ if } \; a \le u \\ \frac{u-4a}{8} < 0, & \text{choosing } \alpha = 0.5 \text{ if } \; a > u \end{cases}.$$

As a result, $\lim_{r\to\infty} \mathrm{EA}(\xi^{(r)};\mathcal{M}') = -\infty$, which contradicts with the fact that $\mathrm{EA}(\xi^{(r)};\mathcal{M}') \ge 0$ for any design $\xi^{(r)}$. Thus,

$$\lim_{r\to\infty} \mathrm{EA}(\xi^{(r)};\mathcal{M}') = \mathrm{EA}(\xi^{\mathrm{Mm}}_{\mathcal{M}'};\mathcal{M}').$$

Since $\ln(\cdot)$ on $[1,\infty)$ is a continuous function,

$$\lim_{r\to\infty} \mathrm{LEA}(\xi^{(r)};\mathcal{M}') = \mathrm{LEA}(\xi^{\mathrm{Mm}}_{\mathcal{M}'};\mathcal{M}').$$

□

S2 Modified Multiplicative Algorithm (Algorithm 2) to Optimize Weights

Description of Algorithm 2.

There are three user-specified parameters δ, Tol, and $MaxIter_2$ in Algorithm 2. Tol is the tolerance of convergence, and we usually set it to be $Tol = 1e{-}15$. $MaxIter_2$ is the maximum number of iterations and we set it to be $MaxIter_2 = 200$ in all numerical examples. The parameter $\delta \in (0,1]$

Algorithm 2 (**Optimal-Weight Procedure**) A Modified Multiplicative Approach.

1: Assign a uniform initial weight vector $\boldsymbol{\lambda}^{(0)} = [\lambda_1^{(0)}, ..., \lambda_n^{(0)}]^\top$, and $k = 0$.

2: **while** *change* $> Tol$ and $k < MaxIter_2$ **do**

3: **for** $i = 1, \dots, n$ **do**

4: Update the weight of design point $\boldsymbol{x}_i$:

$$
\begin{aligned}
\lambda_i^{(k+1)} &= \lambda_i^{(k)} \frac{\left(d_p(\boldsymbol{x}_i, \xi^{\boldsymbol{\lambda}^{(k)}})\right)^\delta}{\sum\limits_{s=1}^n \lambda_s^{(k)} \left(d_p(\boldsymbol{x}_s, \xi^{\boldsymbol{\lambda}^{(k)}})\right)^\delta}, \\
&= \begin{cases}
\lambda_i^{(k)} \dfrac{\left(\sum\limits_{j=1}^m \tilde{\Phi}_0^j(\xi^{\boldsymbol{\lambda}^{(k)}}) w_j(\boldsymbol{x}_i) \boldsymbol{g}_j^\top(\boldsymbol{x}_i) \mathsf{M}_j(\xi^{\boldsymbol{\lambda}^{(k)}}) \boldsymbol{g}_j(\boldsymbol{x}_i)\right)^\delta}{\sum\limits_{s=1}^n \lambda_s^{(k)} \left(\sum\limits_{j=1}^m \tilde{\Phi}_0^j(\xi^{\boldsymbol{\lambda}^{(k)}}) w_j(\boldsymbol{x}_i) \boldsymbol{g}_j^\top(\boldsymbol{x}_i) \mathsf{M}_j(\xi^{\boldsymbol{\lambda}^{(k)}}) \boldsymbol{g}_j(\boldsymbol{x}_i)\right)^\delta}, & p = 0; \\
\lambda_i^{(k)} \dfrac{\left(\sum\limits_{j=1}^m \tilde{\Phi}_p^j(\xi^{\boldsymbol{\lambda}^{(k)}}) w_j(\boldsymbol{x}_i) \left(\mathrm{tr}\left(\mathsf{F}_j(\xi^{\boldsymbol{\lambda}^{(k)}})\right)^p\right)^{1/p-1} \boldsymbol{g}_j^\top(\boldsymbol{x}_i) \mathsf{M}_j(\xi^{\boldsymbol{\lambda}^{(k)}}) \boldsymbol{g}_j(\boldsymbol{x}_i)\right)^\delta}{\sum\limits_{s=1}^n \lambda_s^{(k)} \left(\sum\limits_{j=1}^m \tilde{\Phi}_p^j(\xi^{\boldsymbol{\lambda}^{(k)}}) w_j(\boldsymbol{x}_i) \left(\mathrm{tr}\left(\mathsf{F}_j(\xi^{\boldsymbol{\lambda}^{(k)}})\right)^p\right)^{1/p-1} \boldsymbol{g}_j^\top(\boldsymbol{x}_i) \mathsf{M}_j(\xi^{\boldsymbol{\lambda}^{(k)}}) \boldsymbol{g}_j(\boldsymbol{x}_i)\right)^\delta}, & p > 0.
\end{cases}
\end{aligned} \tag{S2.9}
$$

5: *change* $= \left|\frac{\mathrm{EA}(\boldsymbol{\lambda}^{(k+1)};\mathcal{M}') - \mathrm{EA}(\boldsymbol{\lambda}^{(k)};\mathcal{M}')}{\mathrm{EA}(\boldsymbol{\lambda}^{(k)};\mathcal{M}')}\right|$.

6: $k = k + 1$.

7: **end for**

8: **end while**

plays the same role as in the classical multiplicative algorithm (Silvey et al., 1978), which is to control the speed of the convergence. According to the numerical study by Fellman (1974) and Fiacco and Kortanek (1983), δ is often chosen as 1 for D-optimality, and 0.5 for A- or EI-optimality.

Derivation of (S2.9) is stated as follows. Update the weight of $\boldsymbol{x}_i$ in iteration k with

$$\tilde{\lambda}_i^{(k+1)} = \lambda_i^{(k)} \left(\frac{d_p(\boldsymbol{x}_i, \xi^{\boldsymbol{\lambda}^{(k)}})}{\sum_{t=1}^{n} \lambda_t^{(k)} d_p(\boldsymbol{x}_t, \xi^{\boldsymbol{\lambda}^{(k)}})} \right)^{\delta}, \tag{S2.10}$$

then normalize the weights to ensure the sum 1 condition as

$$\lambda_i^{(k+1)} = \frac{\tilde{\lambda}_i^{(k+1)}}{\sum_{s=1}^{n} \tilde{\lambda}_s^{(k+1)}}. \tag{S2.11}$$

Plugging (S2.10) into (S2.11), we have $\lambda_i^{(k+1)} = \lambda_i^{(k)} \frac{\left(d_p(\boldsymbol{x}_i, \xi^{\boldsymbol{\lambda}^{(k)}})\right)^{\delta}}{\sum_{s=1}^{n} \lambda_s^{(k)} \left(d_p(\boldsymbol{x}_s, \xi^{\boldsymbol{\lambda}^{(k)}})\right)^{\delta}}$.

It is worth pointing out that, occasionally, $\tilde{\Phi}_p^j(\xi^{(r)}) = \left[\Phi_p^{\text{opt}_j}\right]^{-1} \exp\left(\frac{\Phi_p^j(\xi^{(r)})}{\Phi_p^{\text{opt}_j}}\right)$ and $\tilde{\Phi}_p^j(\xi^{\boldsymbol{\lambda}^{(k)}}) = \left[\Phi_p^{\text{opt}_j}\right]^{-1} \exp\left(\frac{\Phi_p^j(\xi^{\boldsymbol{\lambda}^{(k)}})}{\Phi_p^{\text{opt}_j}}\right)$ in (S2.9) of Algorithm 2 and directional derivative $\phi(\boldsymbol{x}, \xi^{(r)})$ in Algorithm 1 can get extreme large and cause overflow, which is a well-recognized issue with the Log-Sum-Exp approximation in the literature. One remedy is to introduce a constant c, and $\exp\left(\frac{\Phi_p^j(\xi^{(r)})}{\Phi_p^{\text{opt}_j}}\right) = \exp(c)\exp\left(\frac{\Phi_p^j(\xi^{(r)})}{\Phi_p^{\text{opt}_j}} - c\right)$. This constant scaling factor $\exp(c)$ is eventually canceled in (S2.9) in Algorithm 2 and does not affect the search for the next design point Algorithm 1. We set $c = \left\lceil \max_j \left(\frac{\Phi_p^j(\xi^{\boldsymbol{\lambda}^{(k)}})}{\Phi_p^{\text{opt}_j}}\right) - 500 \right\rceil$ in Algorithm 2 and $c = \left\lceil \max_j \left(\frac{\Phi_p^j(\xi^{(r)})}{\Phi_p^{\text{opt}_j}}\right) - 500 \right\rceil$ in Algorithm 1 whenever overflow occurs.

Comparison between the proposed Algorithm 2 and existing convex optimization tools.

To show the strength of the proposed Algorithm 2, we use a small and simple example with three design points and two β values. In the following *Example 1*, we compare Algorithm 2 with two existing convex optimization tools, `fmincon` function in MATLAB using interior-point method and the `CVX` toolbox in MATLAB for convex optimization. To solve an optimization problem with the exponential objective function, `CVX` constructed a successive approximation heuristic that approximates the local exponential function with polynomial approximation and solves the approximate model using symmetric primal/dual solvers (Grant et al., 2009).

Example 1. Consider a univariate logistic regression model with the experimental domain $\Omega = [-1, 1]$, basis function $\boldsymbol{g} = [1, x]^\top$ and a parameter space $\mathcal{B} = \{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2\}$ consisting of only two possible regression coefficients $\boldsymbol{\beta}_1 = [-1.4, 2.3]^\top$ and $\boldsymbol{\beta}_2 = [0.5, 1.2]^\top$. The model space is $\mathcal{M} = \{M_1 = (h, \boldsymbol{g}, \boldsymbol{\beta}_1), M_2 = (h, \boldsymbol{g}, \boldsymbol{\beta}_2)\}$, where h is the link function of logistic regression. Given design points $x \in \{-1, 0, 1\}$, all three optimization methods return the same optimal weights,

$$\boldsymbol{\lambda}^* = \{0.3832, 0.2660, 0.3508\}.$$

Table 1 reports the computational times of the three comparison methods. The results clearly show that Algorithm 2 is far more efficient than both `CVX` and `fmincon`. Furthermore, Algorithm 2 boosts the speed of sequential

Algorithm 1 dramatically as finding the optimal weights is done in every iteration of the sequential algorithm.

Table 1: Computational Times (in seconds) of Three Optimization Methods.

CVX	fmincon	Algorithm 2 (Optimal-Weight Procedure)
1.71	0.12	0.03

S3 Connection to Compromise Design

Woods et al. (2006) proposed a compromise design that optimizes the weighted average of certain criteria, where each criterion is based on a potential model from some prior. It means that the compromise design requires a prior distribution $p(M)$ for the model specifications $M \in \mathcal{M}'$. The prior distribution can be as simple as a uniform distribution or other informative distributions.

There can be two different ways to define a compromise design. The first way aims at maximizing a weighted average of the local Φ_p-efficiencies. That is

$$\xi_{\mathcal{M}'}^{\text{eff-com}} = \underset{\xi}{\operatorname{argmax}} \sum_{j=1}^{m} p(M_j)\,\text{eff}_{\Phi_p}(\xi, \xi_{M_j}^{\text{opt}}; M_j),$$

and it is henceforth called the eff-compromise design. Clearly, this averaged local efficiency is not smaller than the reciprocal of LEA$(\xi, \mathcal{M}')$ since

$$[\mathrm{LEA}(\xi; \mathcal{M}')]^{-1} \leq \min_{M \in \mathcal{M}'} \mathrm{eff}_{\Phi_p}(\xi, \xi_M^{\mathrm{opt}}; M) \leq \sum_{j=1}^{m} p(M_j)\, \mathrm{eff}_{\Phi_p}(\xi, \xi_{M_j}^{\mathrm{opt}}; M_j).$$

Thus the compromise design maximizes an upper bound of the worst Φ_p-efficiency. This is not as ideal as LEA$(\xi; \mathcal{M}')$. Minimizing LEA$(\xi; \mathcal{M}')$ simultaneously maximizes a lower and an upper bound of the worst Φ_p-efficiency (see (2.7)), even though the two upper bounds $[\mathrm{LEA}(\xi; \mathcal{M}') - \ln(m)]^{-1}$ and $\sum_{j=1}^{m} p(M_j)\, \mathrm{eff}_{\Phi_p}(\xi, \xi_{M_j}^{\mathrm{opt}}; M_j)$ can be both attainable, depending on the prior distributions.

Another type of compromise design is to minimize the weighted average of local Φ_p-criterion. That is

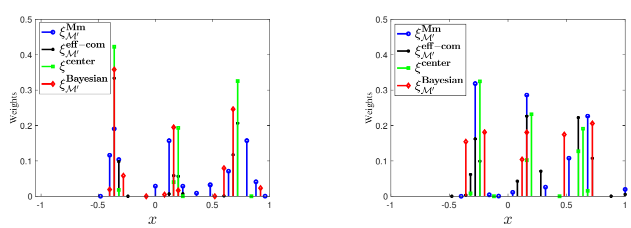
$$\xi_{\mathcal{M}'}^{\Phi_{\mathrm{p}}\text{-com}} = \operatorname*{argmin}_{\xi} \sum_{j=1}^{m} p(M_j) \Phi_p(\xi; M_j),$$

which is henceforth called the Φ_p-compromise design. Such a design criterion is more consistent with the classic Bayesian optimal design. According to Woods et al. (2006) and Atkinson and Woods (2015), the Bayesian optimal design can be considered as a special case of the compromise design, as the former only deals with the uncertainty of the unknown parameters of the GLMs, whereas the compromise design handles all three kinds of uncertainties that are listed previously, including uncertainty of the parameters. We would like to point out that the Φ_p-compromise design can be sensitive to the choice of the prior distribution, especially when the optimal criterion values of different model specifications are very different. On the contrary, LEA$(\xi, \mathcal{M}')$ does not assume any prior distribution and is robust to all the choices of the prior distribution of model specifications.

S4 Additional Tables and Figures in Sections 4.2 - 4.4

Example 3 in Section 4.2

Figure 1 and 2 are the constructed designs and LEA criterion values versus iteration, respectively.



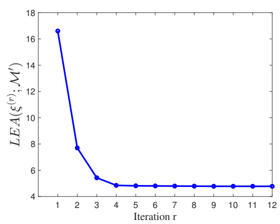
(a) A-optimality (b) D-optimality

Figure 1: Mm-Φ_p Design, Eff-Compromise Design, Centroid Optimal Design and Bayesian Optimal Design

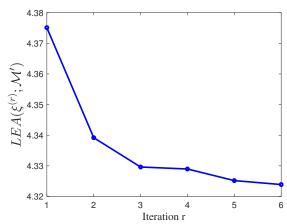
Potato packaging example in Section 4.3

Table 2 includes the estimates of regression coefficients from a preliminary study, and Figure 3 shows the design points of the constructed designs.

Example of exponential growth model in Section 4.4



(a) A-optimality



(b) D-optimality

Figure 2: LEA$(\xi^{(r)}, \mathcal{M}')$ of the r-th iteration in Algorithm 1.

Table 2: Model Space $\mathcal{M}$ of Potato Packing Example

Term	First-Order M_1	With interaction M_2	Second-order M_3
Intercept	-0.28	-1.44	-2.93
x_1	0	0	0
x_2	-0.76	-1.95	-0.52
x_3	-1.15	-2.36	-0.79
x_1x_2		0	0
x_1x_3		0	0
x_2x_3		-2.34	-0.66
x_1^2			0.94
x_2^2			0.79
x_3^2			1.82

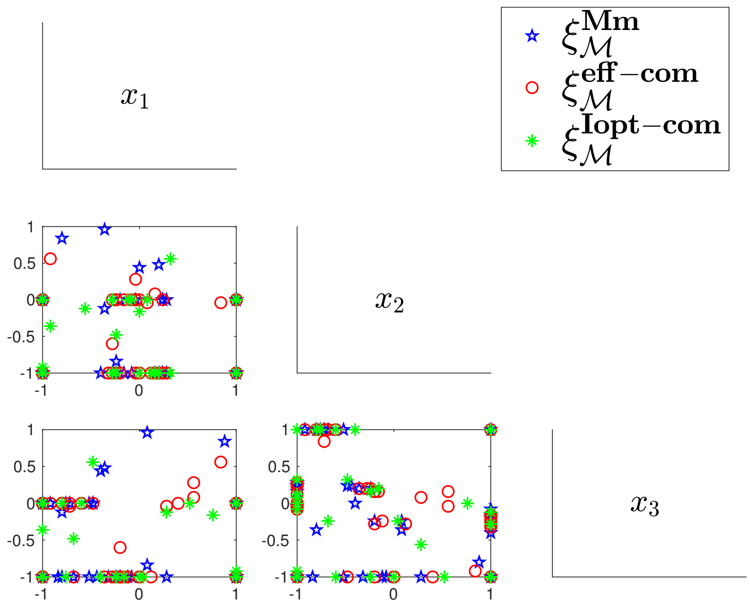


Figure 3: Design Points of Mm-Φ_p Design and Compromise Designs

Table 3 shows the standardized maximin D-optimal designs in (Braess et al., 2007).

Table 3: Standardized maximin D-optimal designs.

B	10	40	100
x_1	0.142	0.037	0.014
λ_1	0.553	0.414	0.336
x_2	0.771	0.193	0.064
λ_2	0.447	0.272	0.193
x_3		0.772	0.156
λ_3		0.314	0.093
x_4			0.287
λ_4			0.137
x_5			0.838
λ_5			0.241

Bibliography

Atkinson, A. C. and Woods, D. C. (2015), "Designs for generalized linear models," *Handbook of Design and Analysis of Experiments*, 471–514.

Braess, D., Dette, H., et al. (2007), "On the number of support points of maximin and Bayesian optimal designs," *The Annals of Statistics*, 35, 772–792.

Fellman, J. (1974), *On the Allocation of Linear Observations*, Societas Scientiarum Fennica.

Fiacco, A. V. and Kortanek, K. O. (eds.) (1983), *A Moment Inequality and Monotonicity of an Algorithm.*

Grant, M., Boyd, S., and Ye, Y. (2009), "cvx users' guide," *online: http://www. stanford. edu/˜ boyd/software. html.*

Li, Y. and Deng, X. (2020), "An efficient algorithm for Elastic I-optimal design of generalized linear models," *Canadian Journal of Statistics.*

Silvey, S. D., Titterington, D. M., and Torsney, B. (1978), "An algorithm for optimal designs on a finite design space," *Communications in Statistics - Theory and Methods*, 14, 1379–1389.

Woods, D. C., Lewis, S. M., Eccleston, J. A., and Russell, K. (2006), "Designs for generalized linear models with several variables and model uncertainty," *Technometrics*, 48, 284–292.

Yang, M. and Stufken, J. (2012), "Identifying Locally Optimal Designs for Nonlinear Models: A Simple Extension with Profound Consequences," *The Annals of Statistics*, 40, 1665–1681.