Application of Particle Swarm Optimization in D-optimal Design

Park, Sungmin

Department of Statistics
Seoul National University

15 MAR 19

Outline

Outline

Locally optimal design for nonlinear model is an optimization problem where analytical formula for the design is rarely available.

This presentation provides,

- a brief introduction to locally optimal design and particle swarm optimization.
- a demonstation of how particle swarm optimization can be successfully implemented in locally optimal design via various examples.

Outline

Research Design

Given an experimental research model, a researcher must decide

1. # of combination levels of indep. variables (# of design points)
2. values for these design points
3. # of subjects to assign at each design points

For example, consider a medical study to test the effect of a new drug. The researcher must decide,

1. # of dose levels to be administered
2. values for each dose level
3. # of patients to be assigned to each dose level

Exact vs. Approximate Design

There are two types of characterization for the design problem:
Exact vs. Approximate

Both specifies 1 # of combination levels and 2 values for these levels.

Two types of design differ in 3,

Exact design specifies the *exact* # of subjects for each design points.

Approximate design specifies the *proportion* of subjects for design points.

In this presentation, we focus on approximate design.

Notation

- $\mathcal{P}(\cdot)$: model
- θ: parameter
- k: # of distinct design points
- $x_i, 1 \leq i \leq k$: design points
- $w_i, 1 \leq i \leq k$: proportion of subjects assigned to design point x_i
- $\xi_k = (x_1, \ldots, x_k, w_1, \ldots, w_k)$: k-point approximate design

Optimal Design, *D*-optimal

A good design is one that produces estimates (usually MLE) with smaller variance or confidence ellipsoid.

Among many optimality criteria, *D*-optimality criterion is widely used.

D-optimality criterion:

$$\text{Given a model } \mathcal{P}(\theta) \text{ and the } \# \text{ of design points } k,$$

$$\xi_k^* \text{ is } D\text{-optimal if } \xi_k^* = \underset{\xi_k}{\operatorname{argmin}} \left| Var\left(\hat{\theta}(\xi)\right) \right|$$

Optimal Design, c-optimal

There is a variant of D-optimal design called c-optimal design.

c-optimality criterion for a function of interest $c(\theta)$:

$$\text{Given a model } \mathcal{P}(\theta) \text{ and the } \# \text{ of design points } k$$

$$\xi_k^* \text{ is } c\text{-optimal if } \xi_k^* = \underset{\xi_k}{\text{argmin}} \left| Var\left[c\left(\hat{\theta}(\xi)\right)\right]\right|,$$

$$\text{with } Var\left[c\left(\hat{\theta}(\xi)\right)\right] \approx \left(\frac{\partial c(\theta)}{\partial \theta}\right) Var\left(\hat{\theta}(\xi)\right) \left(\frac{\partial c(\theta)}{\partial \theta}\right)^T$$

D- and c-optimal design using MLE involves determinant and matrix inversion complicating the optimization problem.
The design solutions are locally optimal because they depend on the choice of parameter θ.

Choice of k

Choice of k is usually pre-specified before the search for optimal design.

In order to estimate p parameters in any model, at least p distinct design points are required.

Carathéodory's theorem provides an upper bound on the number of design points needed for an optimal design.

For many design problems with p parameters,

$$p \leq \text{ Optimal number of distinct design points} \leq \frac{p(p+1)}{2}$$

Start with $k = p$.
If $k = p$ fails to provide an optimal design, try $k = p + 1$ and so on ...

Equivalence Theorem

Equivalence Theorem provides a method to verify whether the approximate design solution is optimal.

Let $f = (f_1, \ldots, f_p)^T$ be a vector of linearly independent real functions on X, whose range is compact in $\mathbb{R}^p$. Let S be any Borel field of subsets of X containing every finite subset of X and ξ a probability measure on S with finite support.
Define $M(\xi) = (m_{ij}(\xi))$ with $m_{ij}(\xi) = \int_X f_i(x) f_j(x) \xi(dx)$ and $d(x;\xi) = f(x)^T \left[M(\xi)\right]^{-1} f(x)$ if $M(\xi)$ is **invertible**. Then, TFAE

1. $\xi^* = \text{argmin}_\xi \left|M(\xi)\right|$
2. $f(x)^T \left[M(\xi^*)\right]^{-1} f(x) \leq p, \forall x \in X$
   with equality holding at x_i^*s in $\xi^* = (x_1^*, \ldots, x_k^*, w_1^*, \ldots, w_k^*)$

Outline

Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a nature-inspired algorithm originating from research in fish and swarm movement behavior.

Benefits of PSO include the ability to find the optimal solution to a complex problem or get close to the optimal solution quickly **without requiring any assumption on the objective function**. $\Rightarrow$ **flexible**

However, the method still lacks a firm theoretical justification to date and is under-utilized in statistical literature.

The idea of PSO is as follows,

1. A number of particles are scattered onto the search domain.
2. Each particle investigates the search domain and shares knowledge with the group.
3. Possible solution is obtained from the group's aggregated knowledge.

PSO Algorithm

Update Equation:

$$v_i^{t+1} = \tau_t v_i^t + \gamma_1 \beta_1 \odot (p_i^t - z_i^t) + \gamma_2 \beta_2 \odot (p_g^t - z_i^t), \quad (1)$$
$$z_i^{t+1} = z_i^t + v_i^{t+1}. \quad (2)$$

- $h(\cdot)$: objective function (fitness) to minimize
- z_i^t: position of the ith particle at time t
- v_i^t: velocity of the ith particle at time t
- p_i^t: $\mathrm{argmin}_{z_i^s, 1 \le s \le t} \{h(z_i^s)\}$, personal best position
- p_g^t: $\mathrm{argmin}_{z_m^s, 1 \le m \le k, 1 \le s \le t} \{h(z_m^s)\}$, global best position
- $\odot$: Hadamard product operator

PSO Algorithm

Update Equation:

$$
\begin{aligned}
v_i^{t+1} &= \tau_t v_i^t + \gamma_1 \beta_1 \odot (p_i^t - z_i^t) + \gamma_2 \beta_2 \odot (p_g^t - z_i^t), \\
z_i^{t+1} &= z_i^t + v_i^{t+1}.
\end{aligned}
$$

- τ_t: inertia wieght at time t, const. or decreasing between (0,1)
- γ_1: cognitive learning parameter
- γ_2: social learning parameter
- β_1, β_2: random vector

In this paper, learning parameter was $\gamma_1 = \gamma_2 = 2$ fixed and components of β_1, β_2 was sampled i.i.d from $U(0, 1)$ at each iteration and particle.

PSO Algorithm

PSO pseudo-code for flock size n (i.e. n particles in the swarm)

(1) Initialize particles

(1.1) Initiate position x_i^0 and velocities v_i^0 for $i = 1, \ldots, n$

(1.2) Calculate the fitness values $h(x_i^0)$ for $i = 1, \ldots, n$

(1.3) Determine the personal best positions $p_i^0 = x_i^0$ and the global position p_g^0 for $i = 1, \ldots, n$

(2) Repeat until stopping criteria are satisfied,

(2.1) Calculate particle velocity according to Eq. (1)

(2.2) Update particle position according to Eq. (2)

(2.3) Project particle back to the search space

(2.4) Calculate the fitness values $h(x_i)$ for v_i for $i = 1, \ldots, n$

(2.5) Update personal and global best positions $p_i, (1 \leq i \leq n)$ and p_g

(3) Output $p_g = \underset{x}{\operatorname{argmin}} \{h(x)\}$ with $gbest = h(p_g)$

Outline

Application of PSO in Optimal Design

Update Equation:

$$
\begin{aligned}
v_i^{t+1} &= \tau_t v_i^t + \gamma_1 \beta_1 \odot (p_i^t - z_i^t) + \gamma_2 \beta_2 \odot (p_g^t - z_i^t), \\
z_i^{t+1} &= z_i^t + v_i^{t+1}.
\end{aligned}
$$

$$
\begin{aligned}
z_i^t \leftarrow \xi_i^t \quad &= (x_{i1}^t, \ldots, x_{ik}^t, w_{i1}^t, \ldots, w_{ik}^t) \\
h(x) \leftarrow g_D(\xi; \theta_0) &= \left| Var\left(\hat{\theta}_{MLE}(\xi)\right)\right| = \left| I(\xi;\theta_0)^{-1}\right|, (D\text{-optimal}) \\
h(x) \leftarrow g_c(\xi; \theta_0) &= \left| Var\left(c\left(\hat{\theta}_{MLE}(\xi)\right)\right)\right|, (c\text{-optimal}) \\
&\approx \left| \left(\left.\frac{\partial c(\theta)}{\partial \theta}\right|_{\theta=\theta_0}\right) I(\xi;\theta_0)^{-1} \left(\left.\frac{\partial c(\theta)}{\partial \theta}\right|_{\theta=\theta_0}\right)^T \right|
\end{aligned}
$$

Application of PSO in Optimal Design

Nominal value θ_0 of θ needs to be specified before running the algorithm.
In most cases, LS estimator is used as the nominal value of θ_0.
Design space, or the search space of x_i also needs to be pre-specified.

Optimality of designs is checked by equivalence theorem for D-optimal designs with invertible Information matrix.

When equivalence theorem is not available, convergence is assumed when observed optimum does not change by more than 10^{-7} deviation from the known optimum.

Outline

Examples: Compartment Model, *D*-optimal

Drug concentration(Y) is modeled as a function($\eta(\cdot, \theta)$) of time(x) with independent normal errors$\left(\varepsilon \sim \mathcal{N}(0, \sigma^2)\right)$.

$$
\begin{aligned}
& Y|X = x \sim \mathcal{N}\left(\eta(x, \theta), \sigma^2\right), \sigma^2 known \\
& \eta(x, \theta) = \theta_3 \left\{\exp(-\theta_2 x) - \exp(-\theta_1 x)\right\}, x > 0 \\
& I(\xi; \theta_0) \propto \sum_{i=1}^{k} \left\{ w_i \left(\left. \frac{\partial \eta(x_i; \theta)}{\partial \theta} \right|_{\theta=\theta_0} \right) \left(\left. \frac{\partial \eta(x_i; \theta)}{\partial \theta} \right|_{\theta=\theta_0} \right)^T \right\} \\
& g_D(\xi; \theta_0) = \left| I(\xi; \theta_0)^{-1} \right|
\end{aligned}
$$

Examples: Compartment Model, D-optimal

D-optimal design:

Find ξ s.t. minimize $g_D(\xi;\theta_0) \Leftrightarrow$ Find ξ s.t. maximize $|I(\xi;\theta_0)|$

- $\theta_0 = (0.05884, 4.298, 21.8)$
- n (flock size) $= 100$
- max iteration $= 100$
- $k = 3$
- $\xi = (x_1, \ldots, x_k, w_1, \ldots, w_k)$,
  $x_i \in [0, 30], w_i \geq 0, \forall i$ and $\sum_{i=1}^{k} w_i = 1$

$\Rightarrow \xi^* = (0.228773, 1.38858, 18.4168, 0.333335, 0.333332, 0.333333)$

Examples: Compartment Model, *D*-optimal

Equivalence plot for *D*-optimal design:

$$\left(\left. \frac{\partial \eta(x;\theta)}{\partial \theta} \right|_{\theta=\theta_0} \right)^T I(\xi^*; \theta_0)^{-1} \left(\left. \frac{\partial \eta(x;\theta)}{\partial \theta} \right|_{\theta=\theta_0} \right) - 3 \leq 0, \forall x \in [0, 30]$$

a.png

Examples: Compartment Model, c-optimal (1)

One of the main question of interest in compartment model is time to max concentration of the drug

$$\begin{aligned} c(\theta) &= \underset{x}{\operatorname{argmax}}\, \eta(x, \theta) \\ &= \frac{\ln \theta_1 - \ln \theta_2}{\theta_1 - \theta_2} \\ \frac{\partial c(\theta)}{\partial \theta} &= \left(\frac{1 - \frac{\theta_2}{\theta_1} - \ln \theta_1 + \ln \theta_2}{(\theta_1 - \theta_2)^2}, \frac{1 - \frac{\theta_1}{\theta_2} + \ln \theta_1 - \ln \theta_2}{(\theta_1 - \theta_2)^2}, 0 \right)^T \\ g_c(\xi; \theta_0) &= \left(\left. \frac{\partial c(\theta)}{\partial \theta} \right|_{\theta=\theta_0} \right)^T I(\xi; \theta_0)^- \left(\left. \frac{\partial c(\theta)}{\partial \theta} \right|_{\theta=\theta_0} \right) \end{aligned}$$

Examples: Compartment Model, c-optimal (1)

$$c\text{-optimal design: find } \xi \text{ s.t. minimize } g_c(\xi;\theta_0)$$

- n (flock size) $= 200$
- max iteration $= 1000$
- $k = 2$
- $\xi = (x_1, \ldots, x_k, w_1, \ldots, w_k)$,
  $x_i \in [0, 10], w_i \geq 0, \forall i$ and $\sum_{i=1}^{k} w_i = 1$

Note that information matrix is singular when $k = 2$.
Instead of using generalized inverse, the author replaced $I(\xi;\theta_0)$ with invertible matrix $I_\epsilon(\xi;\theta_0) = I(\xi;\theta_0) + \epsilon \mathrm{I}_3, \epsilon = 10^{-6}$.

$\Rightarrow \xi^* = (3.56584, 0.179287, 0.393841, 0.606159)$

Examples: Compartment Model, c-optimal (2)

Another question of interest in compartment model is the area under the curve(AUC)

$$
\begin{aligned}
c(\theta) &= \int_0^\infty \eta(x,\theta)dx \\
&= \frac{\theta_3}{\theta_2} - \frac{\theta_3}{\theta_1}
\end{aligned}
$$

$$
\frac{\partial c(\theta)}{\partial \theta} = \left(\frac{\theta_3}{\theta_1^2}, -\frac{\theta_3}{\theta_2^2}, \frac{1}{\theta_2} - \frac{1}{\theta_1}\right)^T
$$

$$
g_c(\xi;\theta_0) = \left(\frac{\partial c(\theta)}{\partial \theta}\bigg|_{\theta=\theta_0}\right)^T I(\xi;\theta_0)^- \left(\frac{\partial c(\theta)}{\partial \theta}\bigg|_{\theta=\theta_0}\right)
$$

Examples: Compartment Model, c-optimal (2)

c-optimal design: find ξ s.t. minimize $g_c(\xi;\theta_0)$

- $k = 2$
- n (flock size) $= 100$
- max iteration $= 1000$
- $\xi = (x_1, \ldots, x_k, w_1, \ldots, w_k)$,
  $x_i \in [0, 20], w_i \geq 0, \forall i$ and $\sum_{i=1}^{k} w_i = 1$

$\Rightarrow \xi^* = (0.23267, 17.634, 0.0135021, 0.986498)$

Examples: Compartment Model, c-optimal (2)

One of the merits of using PSO is that when the # of design points k is over-specified, it can automatically find the optimal design directly.

- $k = 3$, n (flock size) $= 200$, max iteration $= 500$
  $\Rightarrow \xi^* = (0.2327, 17.634, 20.0, 0.0135, 0.9865, 0.0)$ (Simulation)
  $\Rightarrow \xi^* = (0.2337, 17.6269, 17.7176, 0.0135, 0.8983, 0.0882)$ (Paper)
- $k = 3$, n (flock size) $= 200$, max iteration $= 1000$
  $\Rightarrow \xi^* = (0.2327, 17.634, 20.0, 0.0135, 0.9865, 0.0)$ (Simulation)
  $\Rightarrow \xi^* = (0.2332, 17.6336, 17.6626, 0.0135, 0.9535, 0.03296)$ (Paper)

Examples: Compartment Model, c-optimal (2)

Some more simulations,

- $k = 4$, n (flock size) $= 100$, max iteration $= 100$
  $\Rightarrow \xi^* = (0.2337, 17.6126, 17.6464, 20.0, 0.0136, 0.3353, 0.6510, 0.0)$
- $k = 4$, n (flock size) $= 200$, max iteration $= 500$
  $\Rightarrow \xi^* = (0.0, 0.2327, 17.634, 20.0, 0.0, 0.0135, 0.9865, 0.0)$
- $k = 5$, n (flock size) $= 200$, max iteration $= 500$
  $\Rightarrow \xi^* = [0.0, 0.2327, 17.634, 20.0, 20.0, 0, 0.0135, 0.9865, 0.0, 0.0)$

From the simulation, it is observed that when k is too large,

- Neighboring design points converge to optimal design point
- Unnecessary design points converge to the boundary of design space and their weights converge to 0

Examples: Quadratic Logistic Model, *D*-optimal

Consider a binary model taking values 0 or 1 with probability $p(x;\theta)$.

$$
\begin{aligned}
Y|X = x &\sim Ber\left(p(x;\theta)\right) \\
p(x;\theta) &= \frac{\exp[\alpha + \beta(x-\mu)^2]}{1 + \exp[\alpha + \beta(x-\mu)^2]} \\
f(x;\theta) &= \alpha + \beta(x-\mu)^2 \\
\frac{\partial f(x;\theta)}{\partial \theta} &= \left(1, (x_i - \mu)^2, 2\beta(\mu - x_i)\right)^T \\
I(\xi;\theta_0) &= \sum_{i=1}^{k} \left\{ w_i p(x_i;\theta_0)\left(1 - p(x_i;\theta_0)\right) \left(\frac{\partial f(x;\theta)}{\partial \theta}\bigg|_{\theta_0} \right) \left(\frac{\partial f(x;\theta)}{\partial \theta}\bigg|_{\theta_0} \right)^T \right\}
\end{aligned}
$$

Examples: Quadratic Logistic Model, D-optimal

D-optimal design: find ξ s.t. maximize $|I(\xi;\theta_0)|$

- $\theta_0 = (\alpha_0, \beta_0, \mu_0) = (2, 3, 0)$
- n (flock size) $= 128$
- max iteration $= 150$
- $k = 3$
- $\xi = (x_1, \dots, x_k, w_1, \dots, w_k)$,
  $x_i \in [-3, 1], w_i \geq 0, \forall i$ and $\sum_{i=1}^{k} w_i = 1$

$\Rightarrow \xi^* = (-0.726988, 0, 0.726988, 0.333333, 0.333333, 0.333333)$

Outline

Discussion

PSO is a powerful and flexible method for solving optimization problems that can be applied to a variety of problems.

Learning parameter in PSO did not seem to matter much.
Setting $\gamma_1 = \gamma_2 = 2$ worked well in all the problems.

Main problem of PSO is determining the # of iteration and flock size.

- Large flock size will result slow iteration.
- Small flock size will require longer iteration.
  ($\because$ not enough particles to cover the design space)

Reference

- Berger, M. P., & Wong, W. K. (2009). An introduction to optimal designs for social and biomedical research (Vol. 83). John Wiley & Sons.
- Qiu, J., Chen, R. B., Wang, W., & Wong, W. K. (2014). Using animal instincts to design efficient biomedical studies via particle swarm optimization. Swarm and evolutionary computation, 18, 1-10.
- Kiefer, J., & Wolfowitz, J. (1960). The equivalence of two extremum problems. Canadian Journal of Mathematics, 12, 363-366.