1 Lecture 2: Giới thiệu bài toán Partial Optimal Transport

1.1 Hạn chế của Optimal Transport

- $\|\mathbf{r}\|_1$ và $\|\mathbf{c}\|_1$ phải bằng nhau, dẫn đến khó khăn trong một số ứng dụng thực tiễn, khi ta cần một kết nối tối ưu giữa hai "khối lượng" khác nhau, ví dụ như phân loại ảnh hay averaging of neuroimaging data. Việc cố gắng co dãn $\mathbf{r}$ hoặc $\mathbf{c}$ để chúng bằng nhau (nhân thêm hằng số) lại sẽ làm tăng cost.
- Toàn bộ khối lượng $\mathbf{r}$ và $\mathbf{c}$ bắt buộc phải được chuyển $\Rightarrow$ kém linh hoạt.
- Dễ bị outliers trong dữ liệu tác động vào kết quả (do OT muốn vận chuyển toàn bộ $\mathbf{r}$ và $\mathbf{c}$, dẫn đến sẽ có những matching xấu). $\Rightarrow$ Hạn chế lớn nhất của OT đến từ việc yêu cầu cũng như kết quả phải vận chuyển toàn bộ $\mathbf{r}$ và $\mathbf{c}$. Do đó, ta có hướng tiếp cận cho vấn đề trên, chỉ vận chuyển một phần $\mathbf{r}$ vào một phần $\mathbf{c}$. (Partial Optimal Transport).

1.2 Phát biểu bài toán

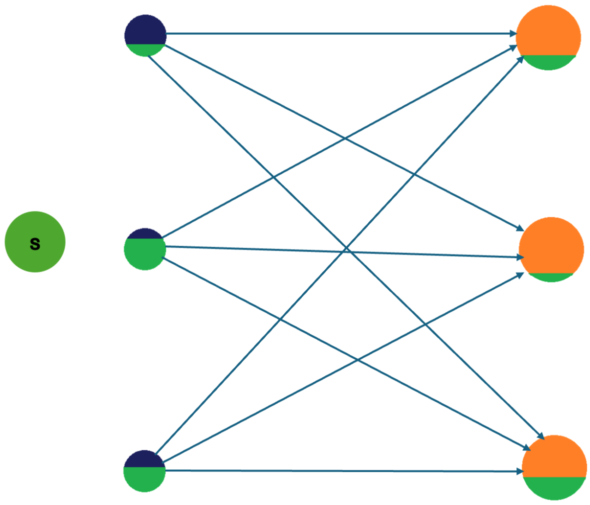
Bài toán Partial Optimal Transport được phát biểu như sau:

$$\mathbf{POT}(\mathbf{r},\mathbf{c},s) = \min\langle \mathbf{C}, \mathbf{X}\rangle \quad \text{với} \quad X \in \mathcal{U}(\mathbf{r},\mathbf{c},s), \tag{1}$$

với $\mathcal{U}(\mathbf{r},\mathbf{c},s)$ là tập hợp

$$\left\{\mathbf{X} \in \mathbb{R}_+^{n\times n} : \mathbf{X}\mathbf{1}_n \le \mathbf{r}, \mathbf{X}^\top \mathbf{1}_n \le \mathbf{c}, \mathbf{1}_n^\top \mathbf{X}\mathbf{1}_n = s\right\},$$

nghĩa là tập hợp các transport map $\mathbf{X}$ có thể chuyển tổng khối lượng s từ $\mathbf{c}$ vào $\mathbf{c}$ và $\mathbf{C} \in \mathbb{R}_+^{n\times n}$ là ma trận cost cho trước.



Hình 1: Vẫn là mô hình 3 vườn, 3 chợ
nhưng với tổng lượng chuối được vận chuyển s.

Nhận xét: Tại sao ta lại cần s trong phát biểu bài toán?

Định nghĩa 1 (Xấp xỉ ε) *Với $\varepsilon \ge 0$, ma trận $\mathbf{X} \in \mathbb{R}_+^{n\times n}$ là một đáp số xấp xỉ ε cho bài toán $\mathbf{POT}(\mathbf{r},\mathbf{c},s)$ if $\mathbf{X} \in \mathcal{U}(\mathbf{r},\mathbf{c},s)$ and*

$$\langle \mathbf{C}, \mathbf{X}\rangle \le \min\langle \mathbf{C}, \mathbf{X}'\rangle + \varepsilon \quad \textit{với} \quad \mathbf{X}' \in \mathcal{U}(\mathbf{r},\mathbf{c},s).$$

Cấu trúc của bài toán có dạng quy hoạch tuyến tính nên các bước phát biểu tiếp theo sẽ dựa vào cách phát biểu bài toán quy hoạch tuyến tính.

Một biểu diễn tương đương khác của bài toán POT: Do $\mathbf{X}\mathbf{1}_n \le \mathbf{r}, \mathbf{X}^\top\mathbf{1}_n \le \mathbf{c}$ nên ta có các vector $\mathbf{p},\mathbf{q} \ge 0$ sao cho $\mathbf{X}\mathbf{1}_n + \mathbf{p} = \mathbf{r}, \mathbf{X}^\top\mathbf{1}_n + \mathbf{q} = \mathbf{c}$.

Lúc này bài toán viết lại thành:

$$\min_{\mathbf{X}\geq 0, \mathbf{p}\geq 0, \mathbf{q}\geq 0} \langle \mathbf{C}, \mathbf{X}\rangle$$
$$\text{với } \mathbf{X}\mathbf{1}_n + \mathbf{p} = \mathbf{r}, \mathbf{X}^\top \mathbf{1}_n + \mathbf{q} = \mathbf{c}, \mathbf{1}_n^\top \mathbf{X}\mathbf{1}_n = s.$$

Nếu ta gộp các điều kiện tuyến tính phía trên thành một điều kiện lớn, ta có thể viết lại bài toán.

Định nghĩa 2 *Với $\mathbf{C} = (\mathbf{c}_1|\mathbf{c}_2|...|\mathbf{c}_n)$, ta ký hiệu vec$(\mathbf{C}) = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_n \end{pmatrix} \in \mathbb{R}_+^{n^2\times 1}$. Tương tự, với $\mathbf{X} = (\mathbf{X}_1|\mathbf{X}_2|...|\mathbf{X}_n)$, ta ký hiệu vec$(\mathbf{X}) = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_n \end{pmatrix} \in \mathbb{R}_+^{n^2\times 1}$. Ở đây, $\mathbf{c}_i$ và $\mathbf{X}_i$'s là các vector cột.*

Bài toán được định nghĩa lại như sau:

$$\min_{\mathbf{x}} \langle \mathbf{d}, \mathbf{x}\rangle \quad \text{với} \quad \mathbf{A}_1\mathbf{x} \leq \mathbf{b}_1, \quad \mathbf{A}_2\mathbf{x} = \mathbf{b}_2 \tag{2}$$

qua việc biến đổi $\mathbf{d} = \text{vec}(\mathbf{C})$ và $\mathbf{x} = (\text{vec}(\mathbf{X}))$ và lựa chọn các giá trị $\mathbf{A}_1, \mathbf{A}_2, \mathbf{b}_1.\mathbf{b}_2$ phù hợp để mô phỏng bài toán POT ban đầu. Ta có thể thấy:

- $\mathbf{A}_1\mathbf{x} \leq \mathbf{b}_1$ sẽ mô phỏng các "constraints": $\mathbf{X}\mathbf{1}_n \leq \mathbf{r}, \mathbf{X}^\top\mathbf{1}_n \leq \mathbf{c}, -\mathbf{X} \leq 0$. (*Câu hỏi:* Vì sao lại là $-\mathbf{X} \leq 0$?). Điều này sẽ tương ứng với lựa chọn $\mathbf{A}_1, \mathbf{b}_1$ sao cho $(\mathbf{A}_1\mathbf{x}) = ((\mathbf{X}\mathbf{1}_n)^\top, (\mathbf{X}^\top\mathbf{1}_n)^\top, -\mathbf{x})^\top$ cũng như $\mathbf{b}_1 = (\mathbf{r}^\top, \mathbf{c}^\top, \mathbf{0}_{n^2}^\top)^\top$. Nói cách khác, ta có phép biến đổi tuyến tính $\mathbf{A}_1$ có dạng:

  $$\mathbf{A}_1 = \begin{pmatrix} \mathbf{A}' \\ \mathbf{1}_{n^2}^\top \end{pmatrix}$$

  với $\mathbf{A}'$ là ma trận edge-incidence ứng với đồ thị lưỡng phân trong bài toán (P)OT. Xem ví du bên dưới để hiểu cách A' được xây dựng.

  Bài tập: Tìm dạng tổng quát của $\mathbf{A}'$.

- $\mathbf{A}_2\mathbf{x} = \mathbf{b}_2$ sẽ mô phỏng "equality constraint": $\mathbf{1}_n^\top\mathbf{X}\mathbf{1}_n = s$. Điều này sẽ tương đương với điều kiện $\mathbf{1}_{n^2}^\top\text{vec}(\mathbf{X}) = \mathbf{1}_{n^2}^\top\mathbf{x} = s$. Như vậy, ta sẽ chọn:

  $$\mathbf{A}_2 = \mathbf{1}_{n^2}^\top, \mathbf{b}_2 = s.$$

  Kiến thức bổ sung: Thật ra, có một mẹo để viết điều kiện $\mathbf{A}_2\mathbf{x} = \mathbf{b}_2$ dưới dạng $\overline{\mathbf{A}_2}\mathbf{x} \leq \overline{\mathbf{b}_2}$. Thật vậy, ta có thể chọn $\overline{\mathbf{A}_2} = [\mathbf{A}_2, -\mathbf{A}_2]$ và $\overline{\mathbf{b}_2} = [\mathbf{b}_2, -\mathbf{b}_2]$. Bằng cách này chúng ta có thể viết bài toán dưới dạng

  $$\max_{\mathbf{x}} \langle -\mathbf{d}, \mathbf{x}\rangle \quad \text{với} \quad \overline{\mathbf{A}}\mathbf{x} \leq \overline{\mathbf{b}}.$$

  Đây cũng là dạng chuẩn của bài toán quy hoạch tuyến tính.

  Bài tập: Chứng minh rằng điều kiện $\mathbf{A}_2\mathbf{x} = \mathbf{b}_2$ tương đương với điều kiện $\overline{\mathbf{A}_2}\mathbf{x} \leq \overline{\mathbf{b}_2}$.

Ví dụ:Với $\mathbf{C} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{pmatrix}$, $\mathbf{r} = (0, 0.8, 0.2)^\top, \mathbf{c} = (0.5, 0.3, 0.2)^\top$:

- $\mathbf{d} = \text{vec}(\mathbf{C}) = (1\ 2\ 1\ 2\ 3\ 2\ 1\ 2\ 1)^\top$
- $\mathbf{x}^T = (\mathbf{X}_{11}\ \mathbf{X}_{21}\ \mathbf{X}_{31}\ \mathbf{X}_{12}\ \mathbf{X}_{22}\ \mathbf{X}_{32}\ \mathbf{X}_{13}\ \mathbf{X}_{23}\ \mathbf{X}_{33})$
- $\mathbf{A}' = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

1.3 Entropic regularization versus quadratic regularization

1.3.1 Entropic regularization

Hàm entropy: Với X là một biến ngẫu nhiên rời rạc với phân phối $P(X)$:
$H(X) = -\sum_{i=1}^{n} \log(P(X_i))P(X_i) = -\langle P(X), \log P(X)\rangle - \mathbb{E}_X[\log(P(X))]$.
Tuy nhiên, trong khi dùng entropic regularization, ta chỉ sử dụng cách xác định công thức, bỏ qua điều kiện X là biến ngẫu nhiên (không cần điều kiện $\sum P(X_i) = 1$ nữa).
Tính chất:

- $H(X)$ đạt giá trị nhỏ nhất (bằng 0) khi và chỉ khi X phân bố đều (xác suất xảy ra bất kì giá trị nào đều bằng nhau).
- H là một hàm α-strong convex. Nghĩa là tồn tại số thực dương α sao cho $\langle \nabla f(y) - \nabla f(x), y - x\rangle \geq \alpha \left\|y - x\right\|^2$.

Câu hỏi: Chứng minh rằng trên $\|X\| \leq D, H(X)$ là một hàm $\frac{1}{D}-$strong convex.

Do tính chất trên, đại lượng này thường được thêm vào hàm tính cost để tạo tính smooth cho hàm.
Khi thêm entropic regularizer này vào cost, ta có phiên bản entropic regularization của bài toán:

$$\min_{\mathbf{x}\geq \mathbf{0}} \{f(x) := \langle \mathbf{d}, \mathbf{x}\rangle + \gamma \langle \mathbf{x}, \log(\mathbf{x})\rangle\} \quad \text{thỏa mãn } \mathbf{Ax} = \mathbf{b},$$

với γ là regularization coefficient tùy chọn, dựa vào việc muốn regularizer mạnh đến đâu. Tính lồi của hàm entropy cũng góp phần tăng tốc độ hội tụ của hàm cost, làm giảm thời gian chạy thuật toán (đã được chứng minh là giảm được từ $\mathcal{O}(n^{3.5})$ với bài toán OT thành $\widetilde{\mathcal{O}}(n^{2.5}/\varepsilon)$[2]).

Tuy vậy, do tính chất $H(X)$ "hướng tới" một transport plan trải đều, ma trận kết quả thường có rất nhiều phần tử khác 0, (vì hạng tử $-x\log x$ tiến về $+\infty$ khi x tiến về 0) vốn có một số vấn đề khi thao tác với chúng ví dụ như numerical instability (sai số), thời gian thao tác.

Trong khi đó, các ma trận sparse (có đa số các phần là 0) giảm thiểu các vấn đề trên, cũng như giảm overfitting đối với bài toán OT. Một cách tiếp cận hướng đến sparse optimal transport là thay thế entropic regularizer thành quadratic regularizer.

1.3.2 Quadratic regularization

Hàm quadratic nhiều biến là hàm số có thể viết dưới dạng $\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$ với $\mathbf{x}$ là vector chứa các biến và $\mathbf{A}$ là một ma trận, $\mathbf{b}$ là một vector, c là một hằng số nào đó.

Exercise: Hãy viết $\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$ dưới dạng tổng các phần tử.

Ví dụ: Đặt $\mathbf{x} = (x_1, ..., x_n)^\top$ thì $(\sum_{i=1}^{n} x_i + 1)^2$ có $\mathbf{A} = \mathbf{1}_{n\times n}, \mathbf{b} = 2 \times \mathbf{1}_n, c = 1$.

Exercise: Hàm trên đạt giá trị nhỏ nhất khi nào? Hãy tìm điểm cực trị bằng các áp dụng giải tích ma trận.

Với một đa thức $\mathbf{P}$ *có n biến, chỉ gồm đơn thức bậc 2, có bao nhiêu ma trận* $\mathbf{A}$ *tương ứng để* P *có thể viết dưới dạng* $\mathbf{x}^\top \mathbf{A}\mathbf{x}$*? Có bao nhiêu ma trận* $\mathbf{A}$ *đối xứng?*

Theorem 1.1 *Cho ma trận* $\mathbf{A}$ *đối xứng* ($\mathbf{A} = \mathbf{A}^\top$) *có kích thước* $n \times n$*. Chứng minh rằng 2 điều sau tương đương:*

- *Với mọi* $\mathbf{x} \in \mathbb{R}^n, \mathbf{x}^\top \mathbf{A}\mathbf{x} \geq 0$
- *Tồn tại ma trận* L *sao cho* $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ *(đọc thêm về Cholesky decomposition)*

Nếu làm mạnh mệnh đề đầu thành với mọi $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0, \mathbf{x}^\top \mathbf{A}\mathbf{x} > 0, \mathbf{L}$ *trong mệnh đề 2 là ma trận khả nghịch. Ma trận* $\mathbf{A}$ *thỏa mãn điều kiện trên gọi là có tính positive semi-definite, thỏa mãn điều kiện mạnh hơn thì có tính positive definite.*

Gợi ý: Thử viết ra $\mathbf{xAx}$ thành dạng đa thức, sau đó dồn biến để viết thành dạng tổng bình phương để tìm $\mathbf{L}$.
- Quadratic regularizer là một hàm bậc hai theo các biến trong ma trận, thường dùng là bình phương của

Frobenius norm: $\|\mathbf{X}\|_F^2 = \sum_{i,j} \mathbf{X}_{ij}^2$.
Sau khi thêm nó vào bài toán POT gốc với hệ số γ, ta được bài toán:

$$\mathbf{QPOT}(\mathbf{r}, \mathbf{c}, s) = \min \langle \mathbf{C}, \mathbf{X} \rangle + \gamma \sum_{i,j} \mathbf{X}_{ij}^2 \quad \text{với} \quad X \in \mathcal{U}(\mathbf{r}, \mathbf{c}, s), \tag{3}$$

Vài lý do để regularizer này hoạt động hiệu quả:

- Cost sẽ tăng đáng kể với các số hạng lớn, nên regularizer này sẽ hạn chế các số hạng lớn, đẩy chúng về gần 0 (trong khi hàm entropy tiến ra vô cùng khi có biến tiến về 0).
- Ở các số hạng có cost tương ứng cao, phần $\langle \mathbf{C}, \mathbf{X} \rangle$ sẽ khiến số hạng đó về gần 0 để hạn chế tổng cost.

Để hiểu rõ hơn về việc các regularizer liên quan thế nào đến việc kết quả có sparse hay không, hãy tham khảo bài báo sau: [1].

Giới thiệu về quadratic programming: Với việc bài toán (3) có các điều kiện tuyến tính tương tự như bài toán OT và ta muốn tìm min của một biểu thức bậc 2 theo các biến ($\mathbf{X}$), bài toán này có dạng quadratic programming.

Bài toán quadratic programming tổng quát:

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^\top Q \mathbf{x} + \mathbf{c}^\top \mathbf{x} \\ \text{Với điều kiện} \quad & A\mathbf{x} \preceq b \end{aligned}$$

1.4 Bài tập về nhà:

1. Cài đặt một quadratic programming solver dùng thư viện cvxpy trong Python.
2. Tính gradient của $\mathbf{x}^\top \mathbf{A} \mathbf{x}$ theo $\mathbf{x}$
3. Tính gradient của $\|\mathbf{x}\|_p$.
4. Trong bài này, giả sử có thể tính được đạo hàm hoặc gradient của các vector theo các biến của nó :
   - Với $\mathbf{x}, \mathbf{y}$ là các vector theo biến t, tính $\frac{\mathrm{d}}{\mathrm{d}t} \langle \mathbf{x}, \mathbf{y} \rangle$
   - (Optional) Với $\mathbf{x}, \mathbf{y}$ là các vector theo các biến $\mathbf{v} = (v_1, ..., v_n)$, tính gradient $\nabla_{\mathbf{v}} \langle \mathbf{x}, \mathbf{y} \rangle$

Tài liệu

[1] M. Blondel, V. Seguy, and A. Rolet. Smooth and sparse optimal transport. In *AISTATS*, pages 880–889, 2018.

[2] Anh Duc Nguyen, Tuan Dung Nguyen, Quang Minh Nguyen, Hoang H. Nguyen, Lam M. Nguyen, and Kim-Chuan Toh. On partial optimal transport: Revising the infeasibility of sinkhorn and efficient gradient methods, 2023.