Class 13

Shikhar Saxena

March 09, 2023

Contents

Recap

Generative Modeling with Optimal Transport

This is the OT (Optimal Transport) Problem:

$$\begin{aligned} d_c(p, q) &= \min_{\pi \in \Pi(p,q)} \langle C, \pi \rangle \\ &= \min_{\pi \in \Pi} \sum_{x,y} c(x, y)\pi(x, y) \\ &\text{where } c(x, y) = \|x - y\|_2 \end{aligned}$$

First Key Idea: **Do entropy regularization**

Entropy Regularized OT

$$d_c^\lambda(p, q) = \min_\pi \langle C, \pi \rangle - \lambda H(\pi)$$

Recall that:

1. $-H$ is strongly convex
2. Given $-H$ is strongly convex, implies d_c^λ is strongly convex
3. Strongly convex also guarantees unique minimizer
4. $\pi(p, q)$ is a compact set

Second Key Idea: **Bring in KL divergence (or formulate OT in that way)**

Consider $k(x, y) = e^{-c(x,y)/\lambda}$ and normalized values $z_\lambda = \sum_x k(x, y)$.

Then we can create a new (artificial) distribution (Gibbs distribution) with probability $p_k^\lambda = \frac{k(x,y)}{z_\lambda}$.

Then we get,

$$D(\pi \| p_k^\lambda) = \frac{1}{\lambda}\ \langle C, \pi \rangle - H(\pi) + \log z_\lambda$$

Remark. *z_λ has no dependence on π (only depends on the cost function c). Thus, minimizing the KL divergence is same as minimizing the entropy regularized OT.*

$$\therefore\ D(\pi \| p_k^\lambda) = \frac{1}{\lambda}\left(\langle C, \pi \rangle - \lambda H(\pi)\right) + \log z_\lambda$$

Third Key Idea (Algorithmic): **Interpret as finding intersection of two convex sets** (Alternating Projection idea)

$$\pi(p,q) = \pi(p) \cap \pi(q)$$

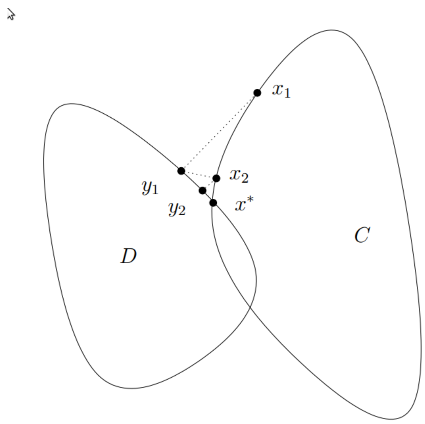

Figure 1: Alternating Projections

The solutions are found as follows:

$$\pi_\lambda^{(2l)} = \text{diag}\left(\frac{p}{\pi_\lambda^{2l+1} 1_m}\right)\pi_\lambda^{2l-1}$$

$$\pi_\lambda^{2l+1} = \text{diag}\left(\frac{q}{1_n^T \pi_\lambda^{(2l)}}\right)\pi_\lambda^{2l}$$

Sinkhorn Algorithm

- Previous algorithm iterated on $\pi_\lambda^{(l)}$.
- We iterate more efficiently here.

Proposition 1. *Let $k \in R^{n \times m}$ with $k_{x,y} = k(x,y)$*
For some $u \in R^n$, $v \in R^m$,

$$\pi_\lambda = \text{diag}(u)\ k\ \ \text{diag}(v)$$

Proof.

$$L(\pi, f, g) = \langle C, \pi \rangle - \lambda H(\pi) + \langle f, p - \pi 1_m \rangle + \langle g, q - \pi^T 1_n \rangle$$
$$\therefore\ \frac{\partial L}{\partial \pi}\bigg|_{(x,y)} = \frac{\lambda}{2} + C(x,y) + \frac{\lambda \log \pi(x,y)}{2} - f_x - g_x = 0$$

Solving for $\pi(x,y)$

$$\implies \lambda \log \pi(x,y) = f_x + g_y - \frac{1}{2}\lambda - \frac{1}{2}\lambda - C(x,y)$$
$$\implies \log \pi(x,y) = \left(\frac{f_x}{\lambda} - \frac{1}{2}\right) + \left(\frac{g_y}{\lambda} - \frac{1}{2}\right) - \frac{C(x,y)}{\lambda}$$
$$\implies \pi(x,y) = e^{f_x/\lambda - 1/2}\ e^{-C(x,y)/\lambda}\ e^{g_y/\lambda - 1/2}$$

This is a subset of **Matrix Scaling** Problems. Scaling algorithms based on the fact that scaling exists. □

Look at $p \otimes q$,

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \otimes \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} p_1 q_1 & p_1 q_2 & \cdots & p_1 q_n \\ p_2 q_1 & p_2 q_2 & \cdots & p_2 q_n \\ \vdots & & & \vdots \\ p_n q_1 & p_n q_2 & \cdots & p_n q_n \end{bmatrix}$$

For matrix, $p \otimes q$, i^{th} row sum is p_i and i^{th} column sum is q_i.

<u>Matrix Scaling</u>: In the discrete case, scaling problem is about finding u and v that scales the rows and columns such that it matches the row and column sum of the given distribution p and q.

In our case,

$$\begin{aligned} p &= \pi 1_m = \text{diag}(u) k v \\ q &= \pi^T 1_n = \text{diag}(v) k^T u \end{aligned} \tag{1}$$

Sinkhorn Algorithm:
Approximate solutions to (1) by initializing

$$u^{(1)} \equiv 1_n,\ \ v^{(1)} \equiv 1_m$$
$$u^{(l+1)} = \frac{p}{k v^{(l)}},\ \ v^{(l+1)} = \frac{q}{k^T u^{(l)}}$$

<u>Claim</u>: With these choices of the iterates u^l and v^l, we obtain primal iterates

Proof. To see this, we run the primal iterates $\pi^{(2l)}$ and $\pi^{(2l+1)}$ with these choices of u^l and v^l in (). Use notation $\tilde{\pi}^{(2l)}$ and $\tilde{\pi}^{(2l+1)}$.

From previous proposition:

$$\tilde{\pi} = \mathrm{diag}(u) \ l \ \mathrm{diag}(v)$$

$$\tilde{\pi}^{(2l)} = \mathrm{diag}(u^{(l+1)}) \ l \ \mathrm{diag}(v^l) \tag{2}$$

$$\tilde{\pi}^{(2l+1)} = \mathrm{diag}(u^{(l+1)}) \ l \ \mathrm{diag}(v^{l+1}) \tag{3}$$

Rearranging terms from (3) ($l = l - 1$),

$$k \ \mathrm{diag}(v^l) = \frac{\tilde{\pi}^{(2l-1)}}{\mathrm{diag}(u^l)} \tag{4}$$

from (2) we have,

$$\tilde{\pi}^{(2l)} = \mathrm{diag}(u^{(l+1)}) \ k \ \mathrm{diag}(v^l)$$

From (4),

$$\begin{aligned}
\tilde{\pi}^{(2l)} &= \mathrm{diag}(u^{(l+1)}) \frac{\tilde{\pi}^{(2l-1)}}{\mathrm{diag}(u^l)} \qquad (5) \\
&= \mathrm{diag}\left(\frac{p}{kv^{(l)}}\right) \frac{\tilde{\pi}^{(2l-1)}}{\mathrm{diag}(u^l)} \\
&= \mathrm{diag}\left(\frac{p}{\mathrm{diag}(u^{(l)} \ k \ v^{(l)}}\right) \tilde{\pi}^{(2l-1)} \\
&= \mathrm{diag}\left(\frac{p}{\tilde{\pi}^{(2l-1)} 1_m}\right) \tilde{\pi}^{(2l-1)}
\end{aligned}$$

Similarly (for odd iterate),

$$\mathrm{diag}(u^{(l+1)})k = \frac{\tilde{\pi}^{(2l)}}{\mathrm{diag}\,(v^l)}$$

Check for $\tilde{\pi}^{(2l+1)}$ (as an exercise).

□

Generative Modeling (using Sinkhorn Algo)

Paper: Learning generative modeling with Sinkhorn Divergence, AISTATS, 2018

The authors took $c(x, y)$ as $f_\phi(x) - f_\phi(y)$ instead of $\|x - y\|_2$.

Algorithm 1: Sinkhorn divergence generative modeling

1. Draw minibatch $x_1, x_2, \dots, x_B \sim p$ and $y_1, y_2, \dots, y_B \sim p_\theta$
2. Approximate $W_1(p, p_\theta) \approx W_1(\hat{p}, \hat{p}_\theta)$

$$\hat{p}(x) = \frac{1}{B} \sum_{i=1}^{B} 1 \; x_i = x, \quad \hat{p}_\theta(x) = \sum_{i=1}^{B} 1 \; y_i = y$$

3. Estimate $W_1(\hat{p}, \hat{p}_\theta)$ using Sinkhorn Algorithm.
4. Compute gradient update

$$\theta = \theta - \eta \nabla_\theta W_1(\hat{p}, \hat{p}_\theta)$$

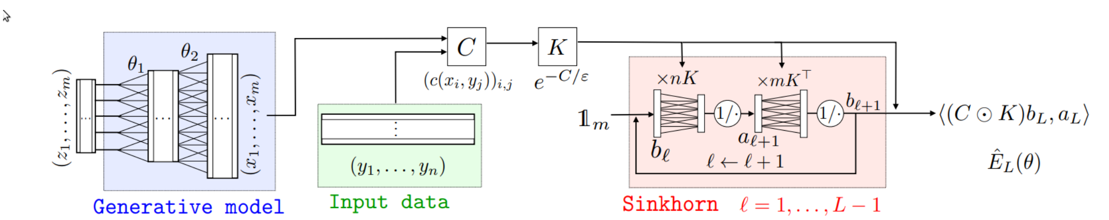

Figure 1: For a given fixed set of samples $(z_1, \dots, z_m)$, and input data $(y_1, \dots, y_n)$, flow diagram for the computation of Sinkhorn loss function $\theta \mapsto \hat{E}_\varepsilon^{(L)}(\theta)$. This function is the one on which automatic differentiation is applied to perform parameter learning. The display shows a simple 2-layer neural network $g_\theta : z \mapsto x$, but this applies to any generative model.

Figure 2: Model Architecture