Optimal transport (OT) / Conditional transport(CT) and their applications

Dongsheng Wang

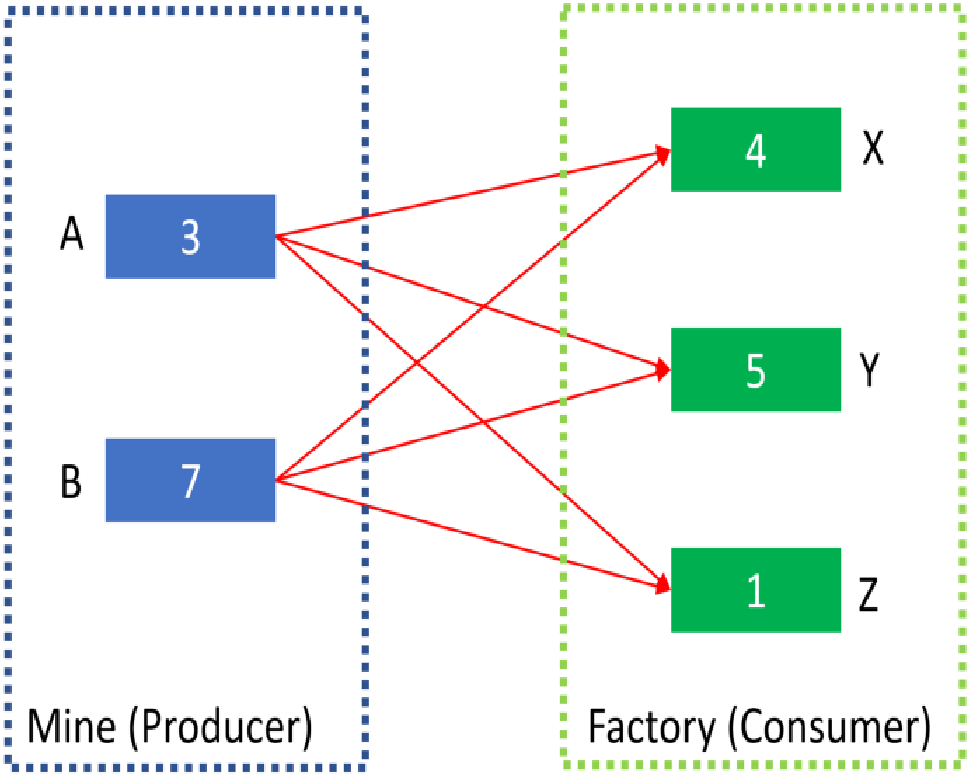
2022.10.12

➢ Outline

- ◆Optimal transport theory
- ◆Sinkhorn distances
- ◆Conditional transport
- ◆Applications and examples

➢ Optimal transport (OT)

◆ Let's start with an simple example



✓ The transport cost (pre-defined)

$$M \in R^{2\times 3} \quad C_{ij} > 0$$

✓ The transport plan (need to-be-optimized)

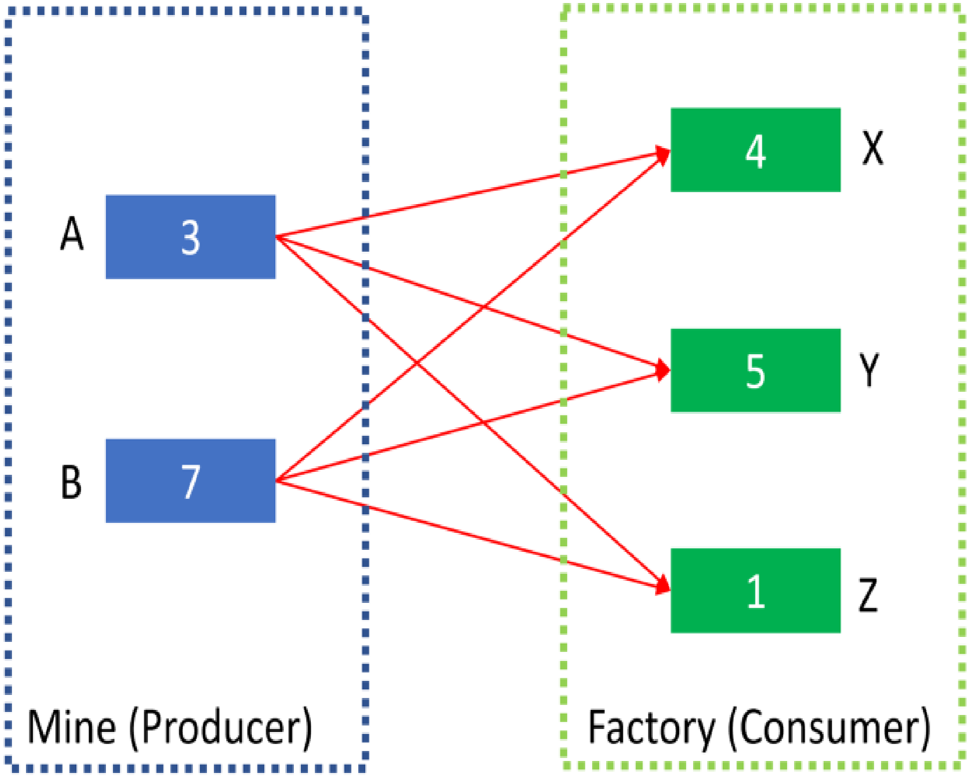
$$P \in R^{2\times 3} \quad \sum_i P_{ij} = [4,5,1] \quad \sum_j P_{ij} = [3,7]^T$$

✓ The total transport cost from the producer to consumer

$$L = \sum_{i\in\{A,B\}} \sum_{j\in\{X,Y,Z\}} P_{ij} M_{ij}$$

➢ Optimal transport (OT)

◆ Let’s start with an simple example



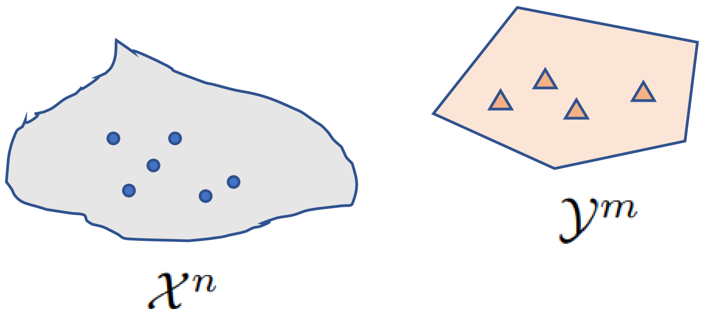
$$d = \min \sum_{ij} P_{ij} C_{ij}$$

$$P_{ij} \geq 0$$

$$\sum_{i} P_{ij} = c_j$$

$$\sum_{j} P_{ij} = r_i$$

➢ Optimal transport (OT)



◆ The OT problem

Two empirical sets:

$$\{x_i\}_{i=1}^n \in \mathcal{X}^n \quad \{y_i\}_{i=1}^m \in \mathcal{Y}^m$$

And the two corresponding weights over samples:

$$\mathbf{p} \in \mathbb{R}_+^n \text{ and } \mathbf{q} \in \mathbb{R}_+^m \text{ where } \sum_{i=1}^n \mathbf{p}_i = \sum_{i=1}^m \mathbf{q}_i = 1$$

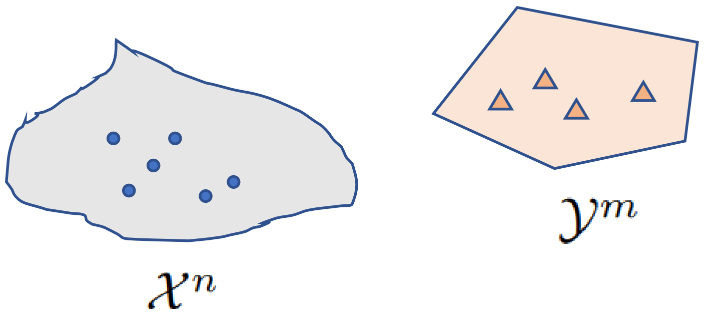
The OT distance between p and q:

$$d_C(p,q) = \min_{\Gamma\in\Pi(\mathbf{p},\mathbf{q})} \langle \Gamma, C\rangle, \quad \Pi(\mathbf{p},\mathbf{q}) = \{\Gamma \in \mathbb{R}_+^{n\times m} | \gamma \mathbf{1}_m = \mathbf{p}, \gamma^T \mathbf{1}_n = \mathbf{q}\}$$

Where C is the transportation cost for each pair, e.g., $C_{ij} = \| x_i - y_j \|$

➢ Optimal transport (OT)

◆ The OT problem



Two empirical sets:

$$\{x_i\}_{i=1}^n \in \mathcal{X}^n \quad \{y_i\}_{i=1}^m \in \mathcal{Y}^m$$

And the two corresponding weights over samples:

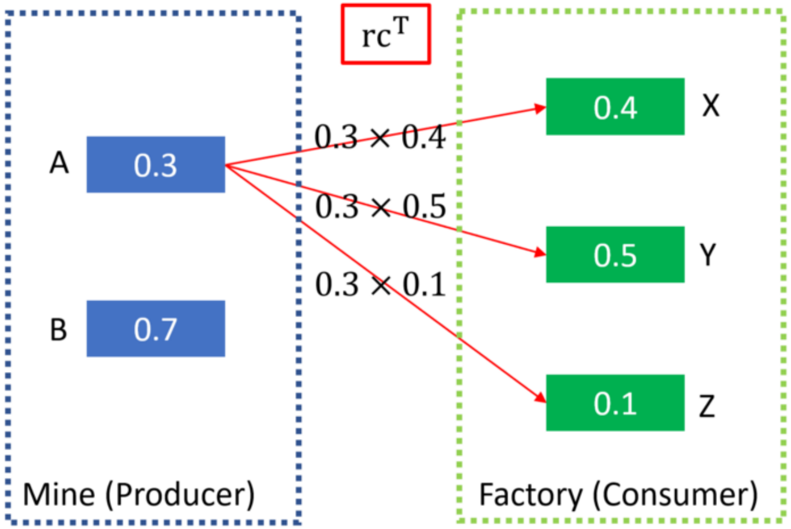
$$\mathbf{p} \in \mathbb{R}_+^n \text{ and } \mathbf{q} \in \mathbb{R}_+^m \text{ where } \sum_{i=1}^n \mathbf{p}_i = \sum_{i=1}^m \mathbf{q}_i = 1$$

The OT distance between p and q:

$$d_C(p,q) = \min_{\Gamma \in \Pi(\mathbf{p},\mathbf{q})} \langle \Gamma, C \rangle, \quad \Pi(\mathbf{p},\mathbf{q}) = \{\Gamma \in \mathbb{R}_+^{n\times m} | \gamma \mathbf{1}_m = \mathbf{p}, \gamma^T \mathbf{1}_n = \mathbf{q}\}$$

The OT problem is a convex problem with the complexity $O(d^3 \log(d))$

➢ Sinkhorn distance



$$d = \min \sum_{i,j} P_{i,j} C_{i,j}$$

$$P_{i,j} \geq 0 \qquad \sum_{j} P_{i,j} = r_i \qquad \sum_{i} P_{i,j} = c_j$$

$$d^* = \min \sum_{i,j} P_{i,j} C_{i,j}$$

$$\cancel{P_{i,j} \geq 0} \qquad \sum_{j} P_{i,j} = r_i \qquad \sum_{i} P_{i,j} = c_j$$

$$KL(\mathrm{P}|\mathrm{rc}^T) \leq \alpha$$

◆What is the rc means

$$KL(\mathrm{P}|\mathrm{rc}^T) = \sum_{i,j} P_{i,j} \log \frac{P_{i,j}}{r_i c_j} = \sum_{i,j} P_{i,j} \log P_{i,j} - \sum_{i,j} P_{i,j} \log r_i - \sum_{i,j} P_{i,j} \log c_j$$

$$h(r) + h(c) - h(P) \leq \alpha$$

$$\hat{d} = \min \sum_{i,j} P_{i,j} C_{i,j} - \frac{1}{\lambda} h(P)$$

$$\sum_{j} P_{i,j} = r_i \qquad \sum_{i} P_{i,j} = c_j$$

➢ Sinkhorn distance

◆ Langrage form of the Sinkhorn distance problem

$$L = \sum_{i,j} P_{i,j} C_{i,j} - \frac{1}{\lambda} h(P) + \sum_{i} m_i \left(\sum_{j} P_{i,j} - r_i \right) + \sum_{j} n_j \left(\sum_{i} P_{i,j} - c_j \right)$$

$$\frac{\partial L}{\partial P_{i,j}} = C_{i,j} + \frac{1}{\lambda} + \frac{1}{\lambda} \log P_{i,j} + m_i + n_j = 0$$

$$P_{i,j} = e^{-\lambda m_i - 0.5} e^{-\lambda C_{i,j}} e^{-\lambda m_j - 0.5}$$

$$P_{i,j} = u_i e^{-\lambda C_{i,j}} v_j$$

$$P = \mathrm{diag(u)} \mathrm{e}^{-\lambda C} \mathrm{diag(v)}$$

➢ Sinkhorn algorithm

◆ The final objective

$$\min_{\Gamma \in \Pi(\mathbf{p},\mathbf{q})} \langle \Gamma, C \rangle - \epsilon H(\Gamma), \text{ where } H(\Gamma) = -\sum_{i,j} \Gamma_{ij} \log \Gamma_{ij}$$

Algorithm 1 Computation of $d_M^\lambda(r,c)$ using Sinkhorn-Knopp's fixed point iteration

Input M, λ, r, c.

I=(r>0); r=r(I); M=M(I,:); K=exp(-λ*M)

Set x=ones(length(r),size(c,2))/length(r);

while x changes **do**

  x=diag(1./r)*K*(c.*(1./(K'*(1./x))))

end while

u=1./x; v=c.*(1./(K'*u))

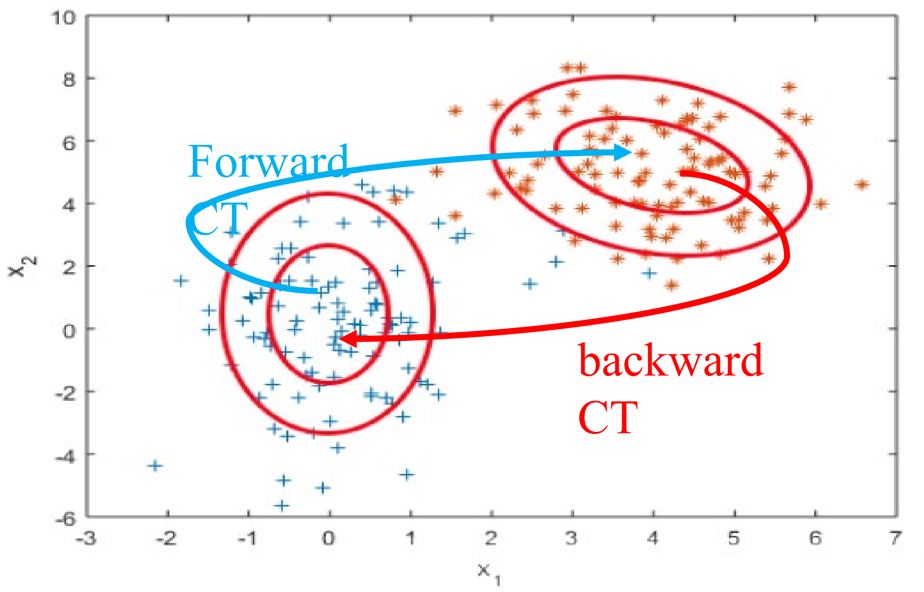
d_M^λ(r,c)=sum(u.*((K.*M)*v))

➢ Conditional transport

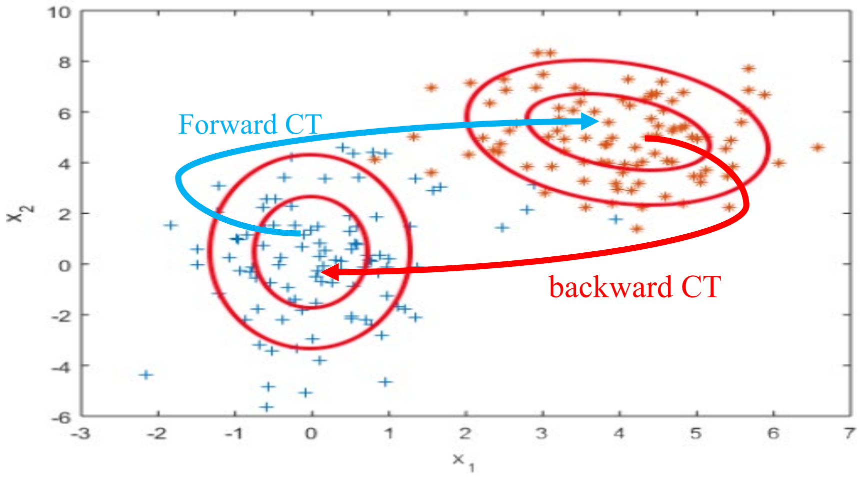
◆Drawbacks of OT

- The OT distance of p and q only considers single-direction transportation, e.g., from p to q
- There is an inner loop in sinkhorn algorithm which is inefficient.

◆CT divergence is defined with a bidirectional distribution-to-distribution transport



➢ Conditional transport



◆ Forward CT is constructed in 3 steps

- Forward navigator

$$\pi(y \mid x) = \frac{e^{-d(x,y)} p_Y(y)}{\int e^{-d(x,y)} p_Y(y) dy}$$

Where, $d(x,y) = d(y,x)$ as a learnable distance function (e.g. $d(x,y) = \frac{(x-y)^2}{2e^{\Phi}}$)

- Cost of forward transport for single point x

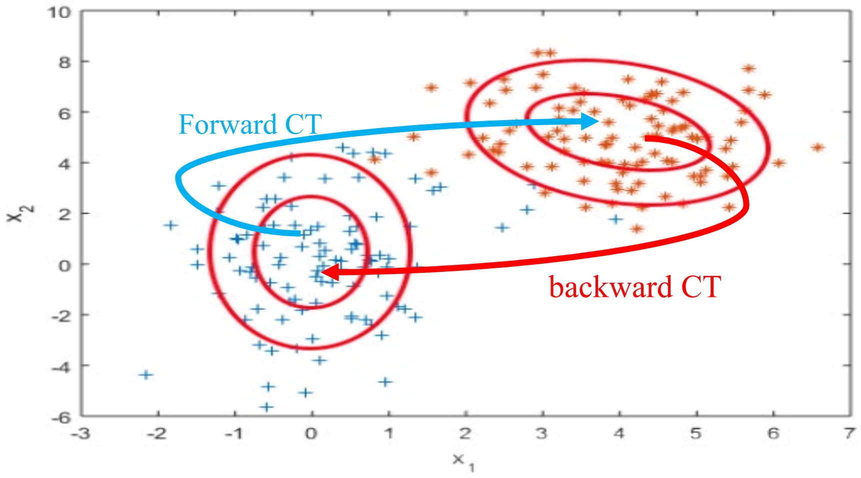
$$\text{cost}_x = \int c(x,y)\pi(y \mid x) dy$$

Where, $c(x,y) = c(y,x) \geq 0$ as the point-to-point transport cost (e.g. $c(x,y) = (x-y)^2$).

- Total cost of forward transport

$$\text{cost} = \int p_X(x) \int c(x,y)\pi(y \mid x) dx dy$$

➢ Conditional transport



✓ Backward CT in the same way

- backward navigator

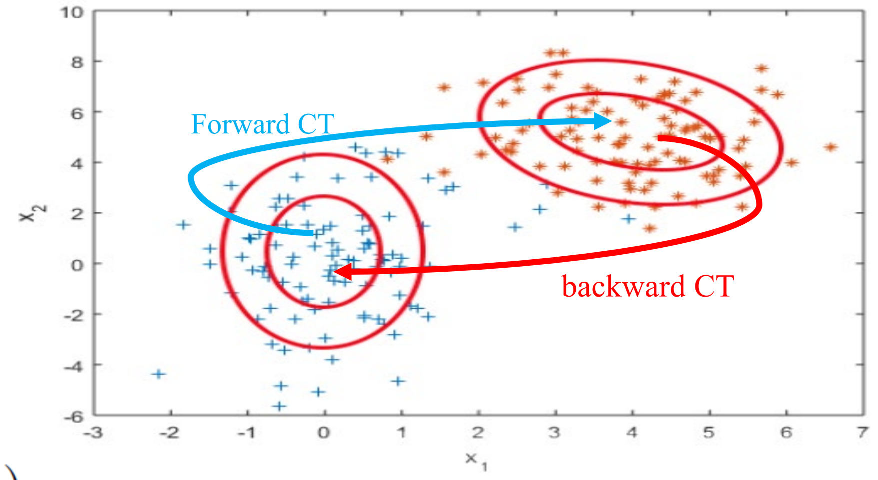
$$\pi(x \mid y) = \frac{e^{-d(x,y)} p_X(x)}{\int e^{-d(x,y)} p_X(x) dx}$$

- Cost of backward transport for single point x

$$\cos t_y = \int c(x,y) \pi(x \mid y) dx$$

- Total cost of backward transport

$$\cos t = \int p_Y(y) \int c(x,y) \pi(x \mid y) dx dy$$



➢ Conditional transport

- CT divergence

$$\mathcal{C}_{\phi,\theta}(\mu,\nu) \overset{\text{def.}}{=} \frac{1}{2}\mathcal{C}_{\phi,\theta}(\mu \to \nu) + \frac{1}{2}\mathcal{C}_{\phi,\theta}(\mu \leftarrow \nu)$$

Where, the forward CT and the backward CT:

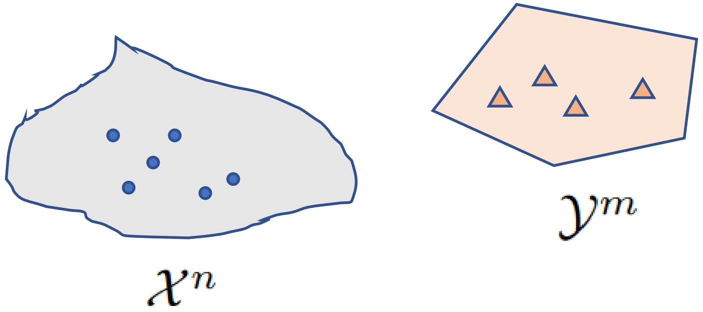
$$\mathcal{C}_{\phi,\theta}(\mu \to \nu) = \mathbb{E}_{x \sim p_X(x)} \mathbb{E}_{y \sim \pi_\phi(y \mid x)}[c(x,y)], \quad \pi_\phi(y \mid x) \overset{\text{def.}}{=} \frac{e^{-d(\mathcal{T}_\phi(x), \mathcal{T}_\phi(y))} p_\theta(y)}{\int e^{-d(\mathcal{T}_\phi(x), \mathcal{T}_\phi(y))} p_\theta(y) dy},$$

$$\mathcal{C}_{\phi,\theta}(\mu \leftarrow \nu) = \mathbb{E}_{y \sim p_\theta(y)} \mathbb{E}_{x \sim \pi_\phi(x \mid y)}[c(x,y)], \quad \pi_\phi(x \mid y) \overset{\text{def.}}{=} \frac{e^{-d(\mathcal{T}_\phi(x), \mathcal{T}_\phi(y))} p_X(x)}{\int e^{-d(\mathcal{T}_\phi(x), \mathcal{T}_\phi(y))} p_X(x) dx},$$

$\mathcal{T}_\phi(\cdot) \in \mathbb{R}^H$, is a neural network based function.

Applications

- Distance function: OT and CT that provide two convenient tools to measure distance between tow empirical distribution (or two sets)
- The most core challenge is to designed the P and Q according to your tasks



➢ Examples of topic model

- Two views of document from empirical distribution over word and topic space

$$p_j = \sum_v \tilde{x}_v \delta_{w_v} \qquad q_j = \sum_k \tilde{\theta}_k \delta_{\beta_k}$$

$\tilde{x} \in \Delta^V$ Normalized Bag-of-Word vector

$w_v \in R^d$ Embedding vector of v-th word

$\tilde{\theta} = f(\tilde{x}) \in \Delta^K$ Normalized topic proportion vector

$\beta_k \in R^d$ Embedding vector of k-th topic

- p and q are two empirical distributions of the same document but with different supports (words and topics)

$$L = \sum_j d_C(p_j, q_j) \qquad C_{vk} = 1 - \cos(w_v, \beta_k)$$

➢ Examples of domain adaptation

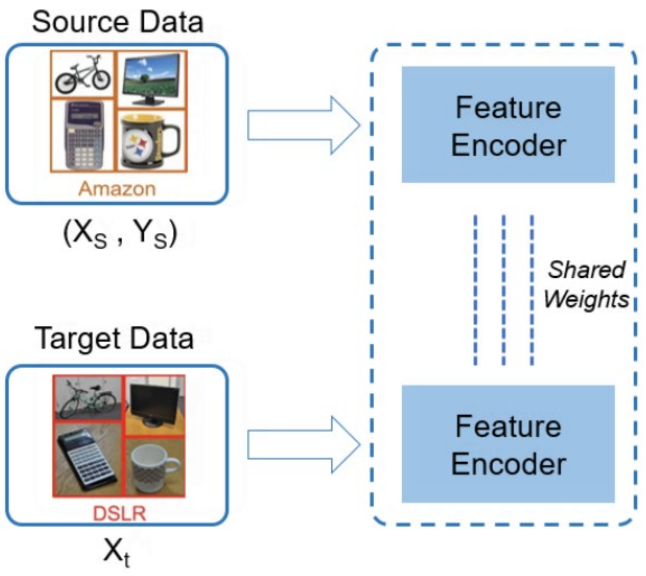
◆The labeled source domain and the unlabeled target domain

$$\{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s} \sim \mathcal{D}_s$$

$$\boldsymbol{f}_i^s = F_{\boldsymbol{\theta}}(\boldsymbol{x}_i^s)$$

$$\{\boldsymbol{x}_j^t\}_{j=1}^{n_t} \sim \mathcal{D}_t^{\boldsymbol{x}}$$

$$\boldsymbol{f}_j^t = F_{\boldsymbol{\theta}}(\boldsymbol{x}_j^t)$$



◆D_s and D_t have the same class label thus share similar class prototypes (centers) in the representation space

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{(\boldsymbol{x}_i^s, y_i^s)\sim\mathcal{D}_s}\left[\sum_{k=1}^{K} -\log p_{ik}^s \mathbf{1}_{\{y_i^s=k\}}\right], \quad p_{ik}^s := \frac{\exp(\boldsymbol{\mu}_k^T \boldsymbol{f}_i^s + b_k)}{\sum_{k'=1}^{K}\exp(\boldsymbol{\mu}_{k'}^T \boldsymbol{f}_i^s + b_{k'})}$$

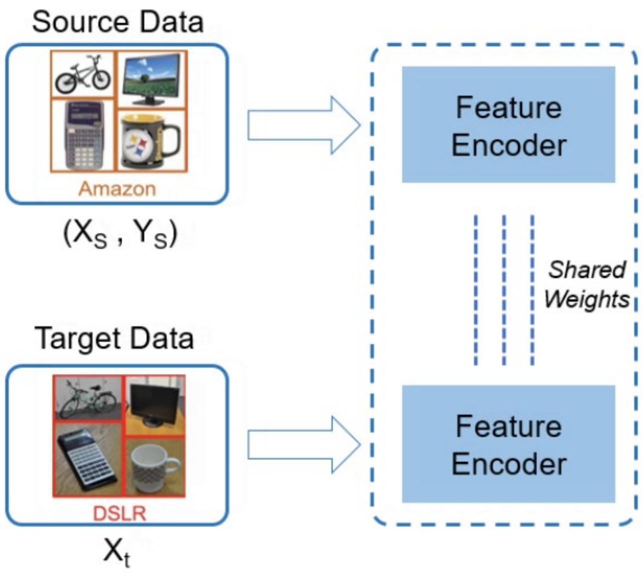
➢ Examples of domain adaptation

◆Two set

$$P = \{f_j^t\}_{j=1}^M \qquad Q = \{u_k\}_{k=1}^K$$

◆CT is employed to finetune the encoder on the target set

$$CT = L_{t\to u} + L_{u\to t}$$



$$\mathcal{L}_{t\to\mu} = \mathbb{E}_{\boldsymbol{x}_j^t\sim\mathcal{D}_t^{\boldsymbol{x}}}\mathbb{E}_{\boldsymbol{\mu}_k\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{\mu}_k\,|\,\boldsymbol{f}_j^t)}\left[c(\boldsymbol{\mu}_k,\boldsymbol{f}_j^t)\right] = \mathbb{E}_{\boldsymbol{x}_j^t\sim\mathcal{D}_t^{\boldsymbol{x}}}\left[\sum_{k=1}^K c(\boldsymbol{\mu}_k,\boldsymbol{f}_j^t)\frac{p(\boldsymbol{\mu}_k)\exp(\boldsymbol{\mu}_k^T\boldsymbol{f}_j^t)}{\sum_{k'=1}^K p(\boldsymbol{\mu}_{k'})\exp(\boldsymbol{\mu}_{k'}^T\boldsymbol{f}_j^t)}\right]$$

$$\mathcal{L}_{\mu\to t} = \mathbb{E}_{\{\boldsymbol{x}_j^t\}_{j=1}^M\sim\mathcal{D}_t^{\boldsymbol{x}}}\mathbb{E}_{\boldsymbol{\mu}_k\sim p(\boldsymbol{\mu}_k)}\mathbb{E}_{\boldsymbol{f}_j^t\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{f}_j^t\,|\,\boldsymbol{\mu}_k)}\left[c(\boldsymbol{\mu}_k,\boldsymbol{f}_j^t)\right]$$
$$= \mathbb{E}_{\{\boldsymbol{x}_j^t\}_{j=1}^M\sim\mathcal{D}_t^{\boldsymbol{x}}}\left[\sum_{k=1}^K p(\boldsymbol{\mu}_k)\sum_{j=1}^M c(\boldsymbol{\mu}_k,\boldsymbol{f}_j^t)\frac{\exp(\boldsymbol{\mu}_k^T\boldsymbol{f}_j^t)}{\sum_{j'=1}^M \exp(\boldsymbol{\mu}_k^T\boldsymbol{f}_{j'}^t)}\right]$$

$$p(\boldsymbol{\mu}_k)^{l+1} = \frac{1}{M}\sum_{j=1}^M \pi_{\boldsymbol{\theta}}^l(\boldsymbol{\mu}_k\,|\,\boldsymbol{f}_j^t), \quad \text{where} \quad \pi_{\boldsymbol{\theta}}^l(\boldsymbol{\mu}_k\,|\,\boldsymbol{f}_j^t) = \frac{p(\boldsymbol{\mu}_k)^l\exp(\boldsymbol{\mu}_k^T\boldsymbol{f}_j^t)}{\sum_{k'=1}^K p(\boldsymbol{\mu}_{k'})^l\exp(\boldsymbol{\mu}_{k'}^T\boldsymbol{f}_j^t)}$$

➢ Examples of imbalanced classification

◆The imbalanced training data and the small balanced meta set

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N} \qquad \mathcal{D}_{\text{meta}} = \{(x_j, y_j)\}_{j=1}^{M} \qquad M \ll N$$

◆Conventional classification

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \ell\left(y_i, f\left(x_i; \boldsymbol{\theta}\right)\right)$$

◆The re-weight classification

$$\boldsymbol{\theta}^*(\boldsymbol{w}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} w_i l_i^{\text{train}}(\theta)$$

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \frac{1}{M} \sum_{j=1}^{M} l_j^{\text{meta}}\left(\boldsymbol{\theta}^*(\boldsymbol{w})\right)$$

- Examples of imbalanced classification

  - Learn w from OT

$$P(\boldsymbol{w}) = \sum\nolimits_{i=1}^{N} w_i \delta_{(x_i, y_i)^{\text{train}}} \qquad Q = \sum\nolimits_{j=1}^{M} \frac{1}{M} \delta_{(x_j, y_j)^{\text{meta}}}$$

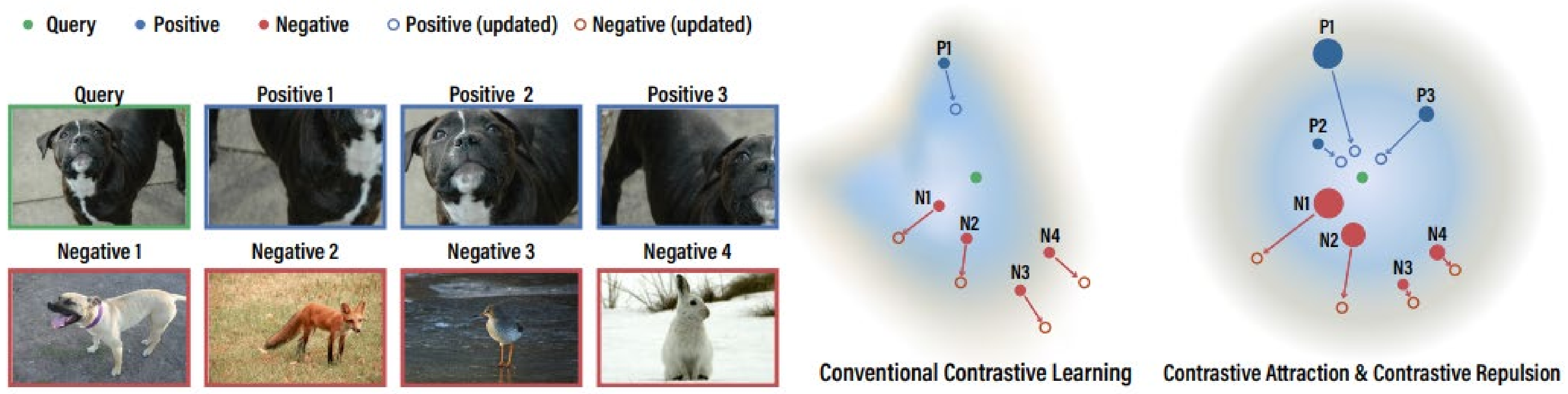
$$\min_{\boldsymbol{w}} \mathrm{OT}(P(\boldsymbol{w}), Q) \stackrel{\text{def.}}{=} \min_{\boldsymbol{w}} \min_{\mathbf{T} \in \Pi(P(\boldsymbol{w}), Q)} \langle \mathbf{T}, \mathbf{C} \rangle$$

  - The cost matrix

$$C_{ij} = \mathrm{d}^{\text{Fea}}(\boldsymbol{z}_i^{\text{train}}, \boldsymbol{z}_j^{\text{meta}}) + \mathrm{d}^{\text{Lab}}(y_i^{\text{train}}, y_j^{\text{meta}})$$

➢ Examples of contrastive learning

◆ The positive samples share close distance while the negative samples share far distance at the representation space



Examples of contrastive learning

- Minimizing the total transport cost of mapping x to its positive and negative sets.

$$L = E_{x\sim p(x)} E_{x^+ \sim \pi_\theta^+(\bullet|x)} [c(f_\theta(x), f_\theta(x^+))] + E_{x\sim p(x)} E_{x^+ \sim \pi_\theta^-(\bullet|x)} [c(f_\theta(x), f_\theta(x^-))]$$

$$\pi_{\boldsymbol{\theta}}^+(\boldsymbol{x}^+ \,|\, \boldsymbol{x}, \boldsymbol{x}_0) := \frac{e^{d_{t^+}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), f_{\boldsymbol{\theta}}(\boldsymbol{x}^+))} p(\boldsymbol{x}^+ \,|\, \boldsymbol{x}_0)}{Q^+(\boldsymbol{x} \,|\, \boldsymbol{x}_0)},$$

$$Q^+(\boldsymbol{x} \,|\, \boldsymbol{x}_0) =: \int e^{d_{t^+}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), f_{\boldsymbol{\theta}}(\boldsymbol{x}^+))} p(\boldsymbol{x}^+ \,|\, \boldsymbol{x}_0) d\boldsymbol{x}^+,$$

$$\pi_{\boldsymbol{\theta}}^-(\boldsymbol{x}^- \,|\, \boldsymbol{x}) := \frac{e^{-d_{t^-}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), f_{\boldsymbol{\theta}}(\boldsymbol{x}^-))} p(\boldsymbol{x}^-)}{Q^-(\boldsymbol{x})},$$

$$Q^-(\boldsymbol{x}) := \int e^{-d_{t^-}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), f_{\boldsymbol{\theta}}(\boldsymbol{x}^-))} p(\boldsymbol{x}^-) d\boldsymbol{x}^-,$$

$$c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2;$$

$$d_{t^+}(\boldsymbol{x}, \boldsymbol{y}) = t^+ \|\boldsymbol{x} - \boldsymbol{y}\|_2^2, \ t^+ \in \mathbb{R}_+;$$

$$d_{t^-}(\boldsymbol{x}, \boldsymbol{y}) = t^- \|\boldsymbol{x} - \boldsymbol{y}\|_2^2, \ t^- \in \mathbb{R}_+.$$

➢ Reference

[1] Zhao, He, et al. "Neural topic model via optimal transport." In ICLR2021

[2] Huynh, Viet, He Zhao, and Dinh Phung. "OTLDA: A geometry-aware optimal transport approach for topic modeling." In NeurIPS2020.

[3] Wang, Dongsheng, et al. "Representing Mixtures of Word Embeddings with Mixtures of Topic Embeddings." *ICLR2022.*

[4] Tanwisuth, Korawat, et al. "A prototype-oriented framework for unsupervised domain adaptation." NeurIPS2021.

[5] Guo, Dandan, et al. "Learning to Re-weight Examples with Optimal Transport for Imbalanced Classification." NeurIPS2022.

[6] BERT-EMD: Many-to-Many Layer Mapping for BERT Compression with Earth Mover's Distance. In ACL2020.

[7] **Learning Prototype-oriented Set Representations for Meta-Learning. In ICLR2022.**

[8] Contrastive Attraction and Contrastive Repulsion for Representation Learning.

[9] Sinkhorn distances: Lightspeed computation of optimal transport. In NeurIPS2013.

[9] From Word Embeddings To Document Distances. In ICML2015.

➢ Links

[1] https://amsword.medium.com/a-simple-introduction-on-sinkhorn-distances-d01a4ef4f085
[2] https://towardsdatascience.com/optimal-transport-a-hidden-gem-that-empowers-todays-machine-learning-2609bbf67e59
[3] http://alexhwilliams.info/itsneuronalblog/2020/10/09/optimal-transport/
[4] https://medium.com/analytics-vidhya/introduction-to-optimal-transport-fd1816d51086
[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6024256/
[6] https://openreview.net/group?id=ICLR.cc/2023/Conference#all-submissions