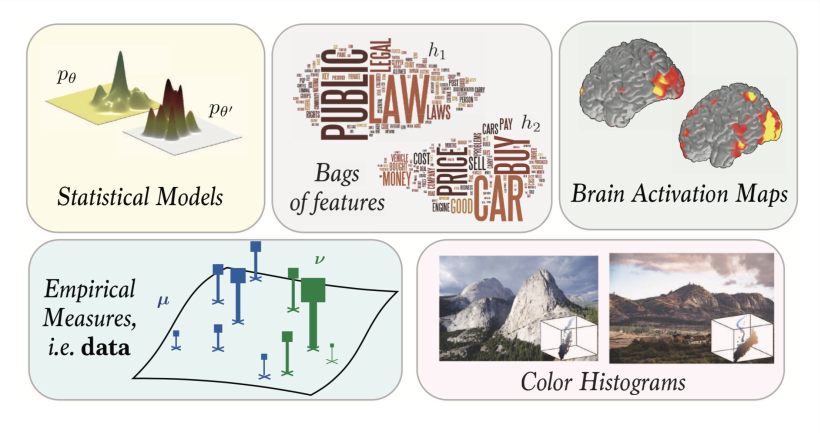
Screenkhorn: Screening Sinkhorn Algorithm for Regularized Optimal Transport

Mokhtar Z. Alaya

Joint work with Maxime Bérar, Gilles Gasso and Alain Rakotomamonjy

Optimal Transport (OT)



[Source image: M. Cuturi (NIPS'17 Tutorial on OT)]

What is it?

A method for comparing probability distributions with the ability to incorporate spatial information.

Regularized Discrete OT Framework: Sinkhorn Divergence

- We consider two discrete probability measures:
  $\boldsymbol{\mu} = \sum_{i=1}^{n} \boldsymbol{\mu}_i \delta_{\boldsymbol{x}_i}$ and $\boldsymbol{\nu} = \sum_{j=1}^{m} \boldsymbol{\nu}_j \delta_{\boldsymbol{x}_j}$.
- We denote their probabilistic couplings set as
  $\boldsymbol{\Pi}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\boldsymbol{P} \in \mathbb{R}_+^{n \times m}, \boldsymbol{P}\mathbf{1}_m = \boldsymbol{\mu}, \boldsymbol{P}^\top \mathbf{1}_n = \boldsymbol{\nu}\}$.
- Cost matrix: $\boldsymbol{C} = (\boldsymbol{C}_{ij}) \in \mathbb{R}_+^{n \times m}$, (e.g., $\boldsymbol{C}_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$).
- Computing an entropic version of OT between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ amounts to solving:

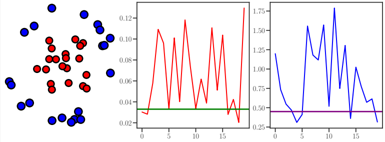
Sinkhorn divergence [Cuturi, 2013]

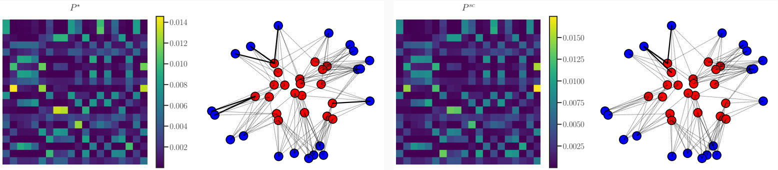
$$\mathcal{S}_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{P} \in \boldsymbol{\Pi}(\boldsymbol{\mu}, \boldsymbol{\nu})} \{\langle \boldsymbol{C}, \boldsymbol{P} \rangle - \eta H(\boldsymbol{P})\}.$$

- Negative entropy $H(\boldsymbol{P}) = -\sum_{i,j} \boldsymbol{P}_{ij} \log(\boldsymbol{P}_{ij})$ and $\eta > 0$ is a regularization parameter.

Screened Dual of Sinkhorn Divergence: Motivation

- OT plan presents a large number of neglectable values [Blondel et al., 2018].
- **Static screening test** in Lasso [Ghaoui et al., 2010].
- We define the convex set $\mathcal{C}^r_\alpha = \{w \in \mathbb{R}^r : e^{w_i} \geq \alpha\}$, for $\alpha > 0$.





- Identify these indices and fixed at the thresholds before solving the problem. $\rightarrow$ Reduce the scale of the optim. procedure.

Static Screening Test: Approximate Dual of $\mathcal{S}_\eta(\boldsymbol{\mu}, \boldsymbol{\nu})$

- Based on this idea, we define a so-called *approximate dual of Sinkhorn divergence*

Approximate dual of Sinkhorn divergence

$$\mathcal{S}_\eta^{\mathrm{ad}}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{u} \in \mathcal{C}^n_{\frac{\varepsilon}{\kappa}}, \boldsymbol{v} \in \mathcal{C}^m_{\varepsilon\kappa}} \Big\{ \Psi_\kappa(\boldsymbol{u}, \boldsymbol{v}) := \mathbf{1}_n^\top B(\boldsymbol{u}, \boldsymbol{v}) \mathbf{1}_m - \langle \kappa \boldsymbol{u}, \boldsymbol{\mu} \rangle - \langle \frac{\boldsymbol{v}}{\kappa}, \boldsymbol{\nu} \rangle \Big\}.$$

Screening with a Fixed Number Budget of Points

- We denote by $n_b \in \{1, \ldots, n\}$ and $m_b \in \{1, \ldots, m\}$ the number of points that are going to be optimized in $\mathcal{S}_\eta^{\text{ad}}(\boldsymbol{\mu}, \boldsymbol{\nu})$.
- Let $\boldsymbol{\xi} \in \mathbb{R}^n$ and $\boldsymbol{\zeta} \in \mathbb{R}^m$ to be the ordered decreasing vectors of $\boldsymbol{\mu} \oslash r(\boldsymbol{K})$ and $\boldsymbol{\nu} \oslash c(\boldsymbol{K})$ respectively.
- To keep only n_b-budget and m_b-budget of points, the parameters κ and ε satisfy $\frac{\varepsilon^2}{\kappa} = \boldsymbol{\xi}_{n_b}$ and $\varepsilon^2 \kappa = \boldsymbol{\zeta}_{m_b}$. Then

$$\varepsilon = (\boldsymbol{\xi}_{n_b} \boldsymbol{\zeta}_{m_b})^{1/4} \text{ and } \kappa = \sqrt{\frac{\boldsymbol{\zeta}_{m_b}}{\boldsymbol{\xi}_{n_b}}}.$$

Screening with a Fixed Number Budget of Points

- We can restrict the variables in $\mathcal{S}_{\eta}^{\mathrm{ad}}(\boldsymbol{\mu}, \boldsymbol{\nu})$ to variables in $I_{\varepsilon,\kappa}$ and $J_{\varepsilon,\kappa}$ where $I_{\varepsilon,\kappa} = \left\{i = 1, \ldots, n : \boldsymbol{\mu}_i \geq \frac{\varepsilon^2}{\kappa} r_i(\boldsymbol{K})\right\}$ and $J_{\varepsilon,\kappa} = \left\{j = 1, \ldots, m : \boldsymbol{\nu}_j \geq \kappa\varepsilon^2 c_j(\boldsymbol{K})\right\}$.
- This boils down to restricting the constraints feasibility $\mathcal{C}^n_{\frac{\varepsilon}{\kappa}} \cap \mathcal{C}^m_{\varepsilon\kappa}$ to the *screened domain* defined by $\mathcal{U}^{\mathrm{sc}} \cap \mathcal{V}^{\mathrm{sc}}$, where

Screened feasibility domain

$$\mathcal{U}^{\mathrm{sc}} = \{\boldsymbol{u} \in \mathbb{R}^{n_b} : e^{\boldsymbol{u}_i} \geq \frac{\varepsilon}{\kappa}\} \text{ and } \mathcal{V}^{\mathrm{sc}} = \{\boldsymbol{v} \in \mathbb{R}^{m_b} : e^{\boldsymbol{v}_j} \geq \varepsilon\kappa\}.$$

Screenkhorn

Algorithm 1: Screenkhorn($\boldsymbol{C}, \eta, \boldsymbol{\mu}, \boldsymbol{\nu}, n_b, m_b$)

Step 1: Screening

$\boldsymbol{K} \leftarrow e^{-\boldsymbol{C}/\eta}$;

$\boldsymbol{\xi} \leftarrow \mathtt{sort}(\boldsymbol{\mu} \oslash r(\boldsymbol{K}))$, $\boldsymbol{\zeta} \leftarrow \mathtt{sort}(\boldsymbol{\nu} \oslash c(\boldsymbol{K}))$; //(decreasing order)

$\varepsilon \leftarrow (\boldsymbol{\xi}_{n_b}\boldsymbol{\zeta}_{m_b})^{1/4}$, $\kappa \leftarrow \sqrt{\boldsymbol{\zeta}_{m_b}/\boldsymbol{\xi}_{n_b}}$;

$I_{\varepsilon,\kappa} \leftarrow \{i = 1, \ldots, n : \boldsymbol{\mu}_i \geq \varepsilon^2 \kappa^{-1} r_i(\boldsymbol{K})\}$;

$J_{\varepsilon,\kappa} \leftarrow \{j = 1, \ldots, m : \boldsymbol{\nu}_j \geq \varepsilon^2 \kappa c_j(\boldsymbol{K})\}$;

$\boldsymbol{K}_{\min} = \min_{I_{\varepsilon,\kappa}, J_{\varepsilon,\kappa}} \boldsymbol{K}_{ij}$;

$\underline{\boldsymbol{\mu}} \leftarrow \min_{i \in I_{\varepsilon,\kappa}} \boldsymbol{\mu}_i$, $\bar{\boldsymbol{\mu}} \leftarrow \max_{i \in I_{\varepsilon,\kappa}} \boldsymbol{\mu}_i$; $\underline{\boldsymbol{\nu}} \leftarrow \min_{j \in J_{\varepsilon,\kappa}} \boldsymbol{\mu}_i$, $\bar{\boldsymbol{\nu}} \leftarrow \max_{j \in J_{\varepsilon,\kappa}} \boldsymbol{\mu}_i$;

$\underline{\boldsymbol{u}} \leftarrow \log\left(\frac{\varepsilon}{\kappa} \vee \frac{\underline{\boldsymbol{\mu}}}{\varepsilon(m-m_b)+\varepsilon \vee \frac{\bar{\boldsymbol{\nu}}}{n\varepsilon\kappa \boldsymbol{K}_{\min}} m_b}\right)$, $\bar{\boldsymbol{u}} \leftarrow \log\left(\frac{\varepsilon}{\kappa} \vee \frac{\bar{\boldsymbol{\mu}}}{m\varepsilon \boldsymbol{K}_{\min}}\right)$;

$\underline{\boldsymbol{v}} \leftarrow \log\left(\varepsilon\kappa \vee \frac{\underline{\boldsymbol{\nu}}}{\varepsilon(n-n_b)+\varepsilon \vee \frac{\kappa\bar{\boldsymbol{\mu}}}{m\varepsilon \boldsymbol{K}_{\min}} n_b}\right)$, $\bar{\boldsymbol{v}} \leftarrow \log\left(\varepsilon\kappa \vee \frac{\bar{\boldsymbol{\nu}}}{n\varepsilon \boldsymbol{K}_{\min}}\right)$;

$\bar{\boldsymbol{\theta}} \leftarrow \mathtt{stack}(\bar{\boldsymbol{u}}\mathbf{1}_{n_b}, \bar{\boldsymbol{v}}\mathbf{1}_{m_b})$, $\underline{\boldsymbol{\theta}} \leftarrow \mathtt{stack}(\underline{\boldsymbol{u}}\mathbf{1}_{n_b}, \underline{\boldsymbol{v}}\mathbf{1}_{m_b})$;

Step 2: L-BFGS-B (SciPy Library)

$\boldsymbol{u}^{(0)} \leftarrow \log(\varepsilon\kappa^{-1})\mathbf{1}_{n_b}$, $\boldsymbol{v}^{(0)} \leftarrow \log(\varepsilon\kappa)\mathbf{1}_{m_b}$;

$\boldsymbol{\theta}^{(0)} \leftarrow \mathtt{stack}(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$;

$\boldsymbol{\theta} \leftarrow$ L-BFGS-B$(\boldsymbol{\theta}^{(0)}, \underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}})$;

$\boldsymbol{\theta}_u \leftarrow (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{n_b})^\top$;

$\boldsymbol{\theta}_v \leftarrow (\boldsymbol{\theta}_{n_b+1}, \ldots, \boldsymbol{\theta}_{n_b+m_b})^\top$;

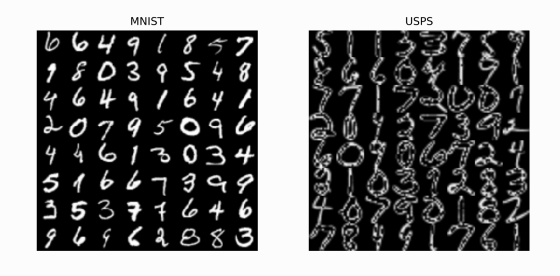
$\boldsymbol{u}^{sc}_i \leftarrow (\boldsymbol{\theta}_u)_i$ if $i \in I_{\varepsilon,\kappa}$ and $\boldsymbol{u}^{sc}_i \leftarrow \log(\varepsilon\kappa^{-1})$ if $i \in I^{\complement}_{\varepsilon,\kappa}$;

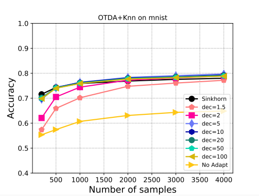
$\boldsymbol{v}^{sc}_j \leftarrow (\boldsymbol{\theta}_v)_j$ if $j \in J_{\varepsilon,\kappa}$ and $\boldsymbol{v}^{sc}_j \leftarrow \log(\varepsilon\kappa)$ if $j \in J^{\complement}_{\varepsilon,\kappa}$;

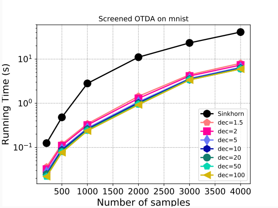
return $B(\boldsymbol{u}^{sc}, \boldsymbol{v}^{sc})$.

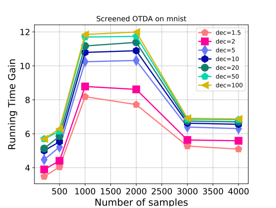
Integrating Screenkhorn in ML Pipeline

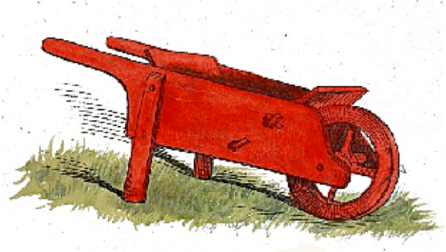
Optimal Transport Domain Adaptation (OTDA) [Courty et al., 2017]: MNIST (source) to USPS (target).

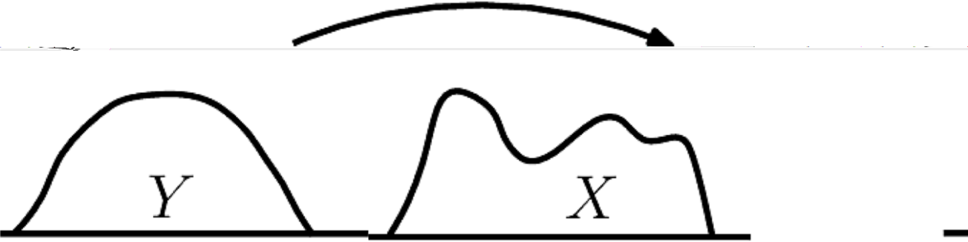











Y
X