
LABORATOIRE
HUBERT CURIEN
UMR • CNRS • 5516 • SAINT-ETIENNE

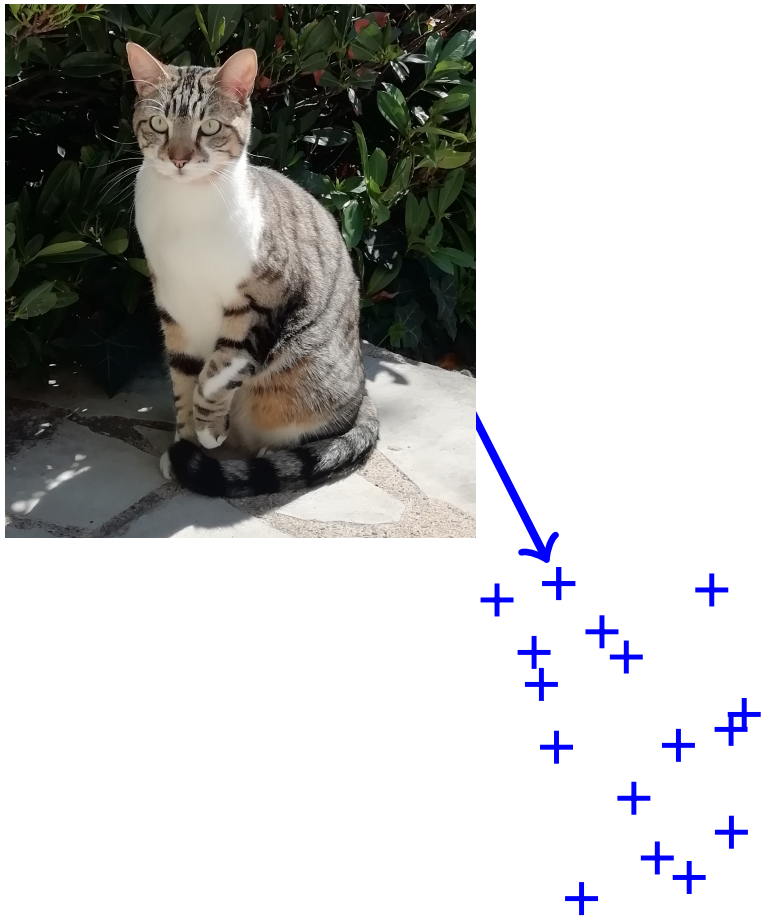
Contributions to Optimal Transport for Machine Learning: Ground Metric and Generalized Framework

Thesis defense of Tanguy KERDONCUFF

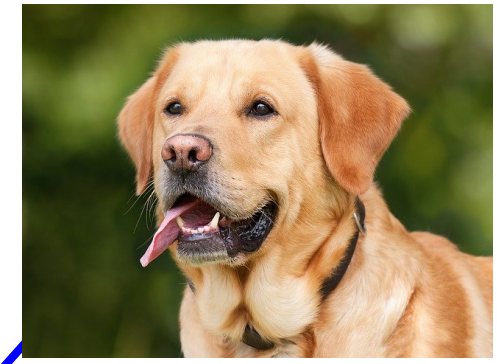
Élisa FROMONT Professeure, Université de Renne 1 Présidente
Marianne CLAUSEL Professeure, Université de Lorraine Rapporteuse
Nicolas COURTY Professeur, Université Bretagne Sud Rapporteur
Marc SEBBAN Professeur, Université de Saint-Étienne Directeur
Rémi EMONET Maître de conférences, Université de Saint-Étienne Co-encadrant

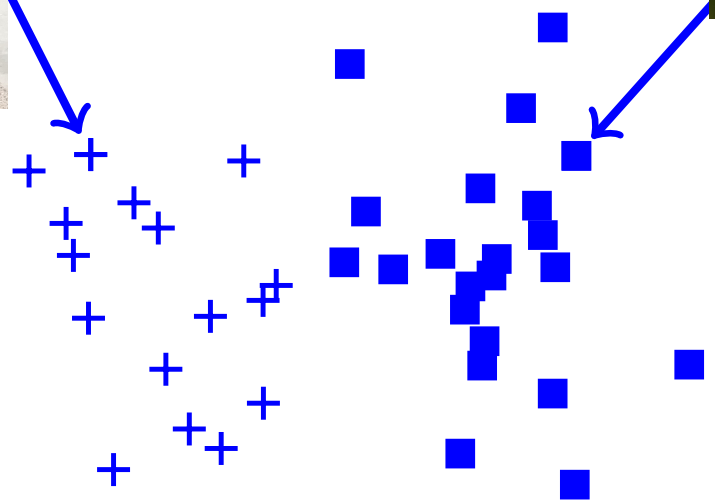
November 16, 2021

Basic Machine Learning examples

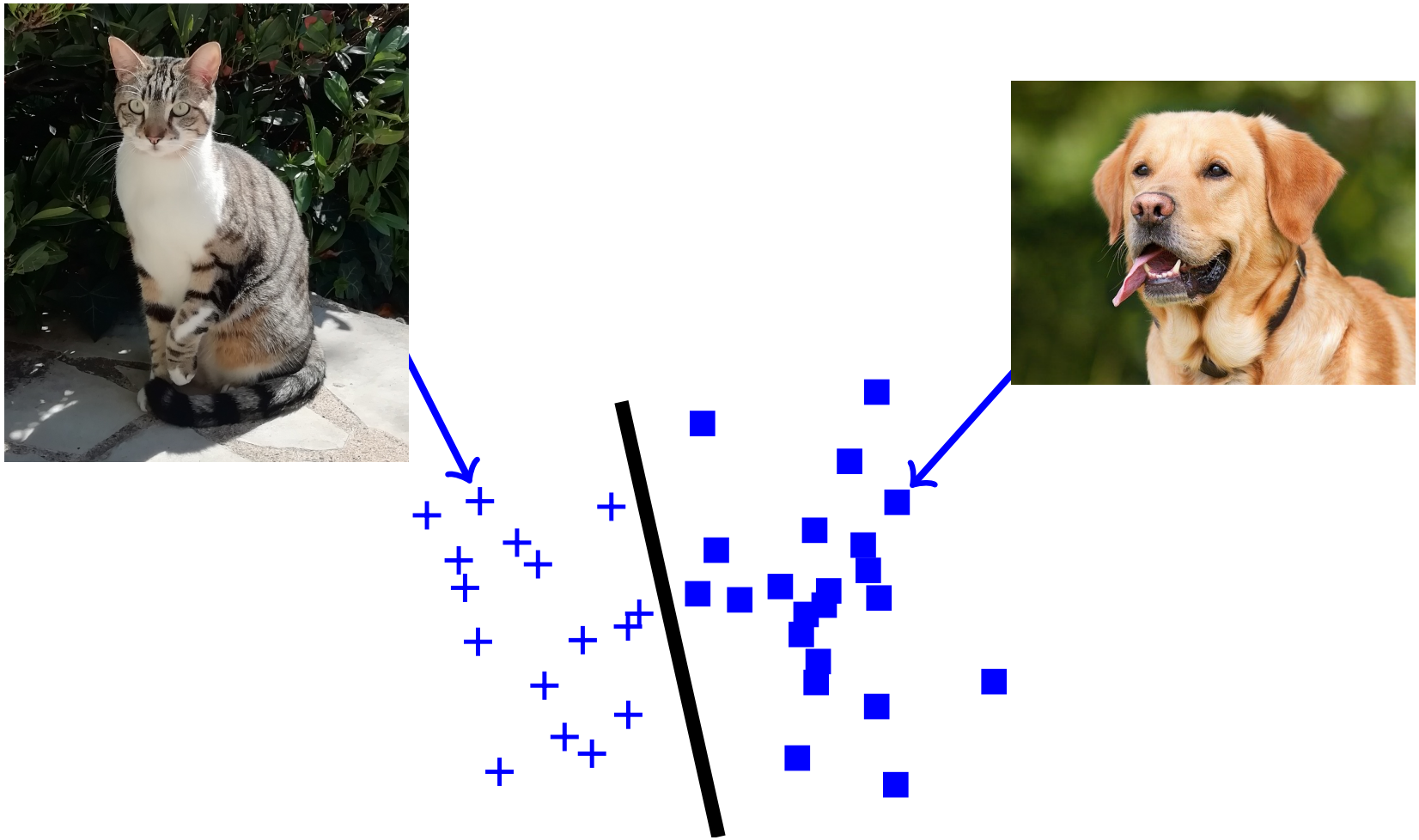


Basic Machine Learning examples

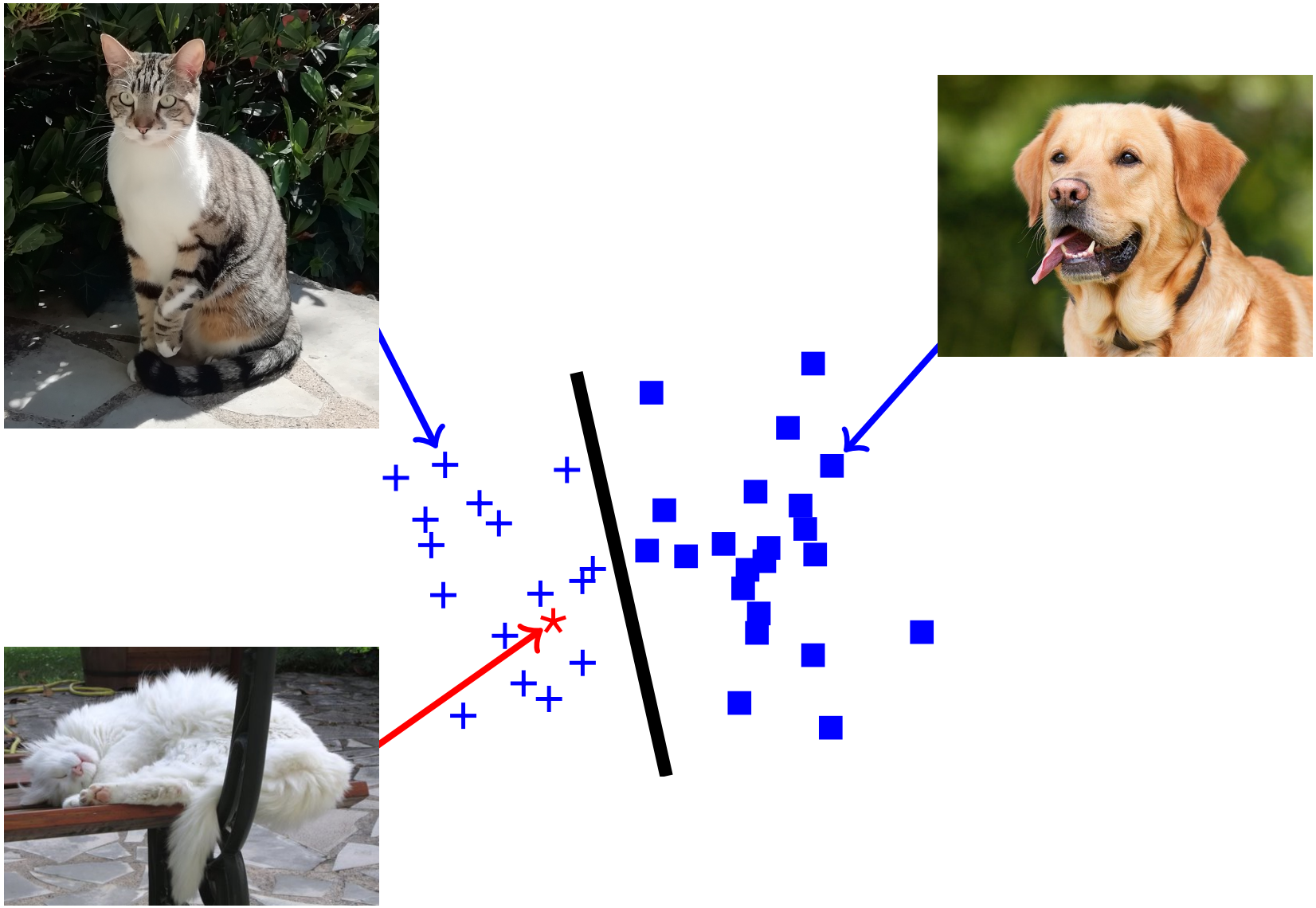




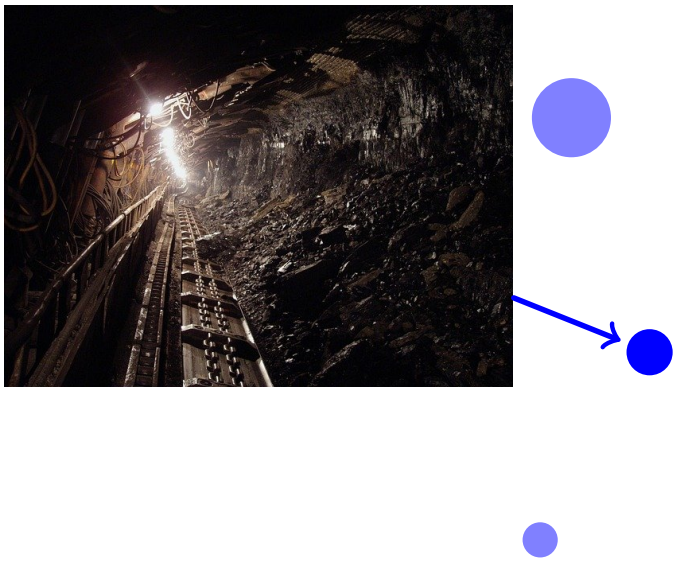
Basic Machine Learning examples



Basic Machine Learning examples

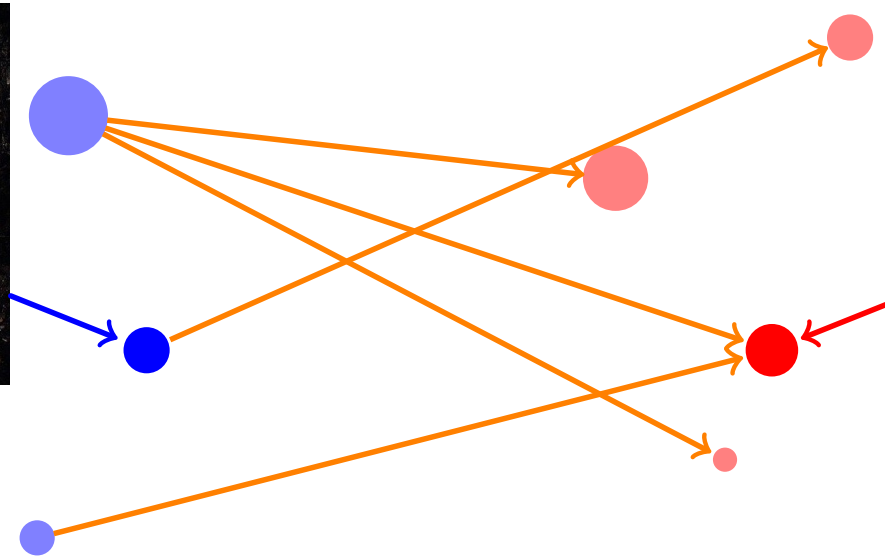


Intuition of the Optimal Transport

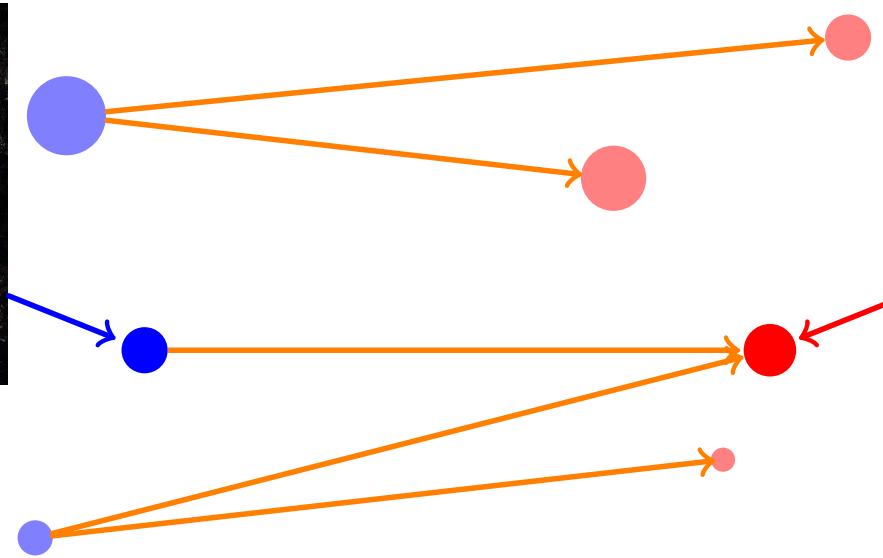


Intuition of the Optimal Transport

Intuition of the Optimal Transport

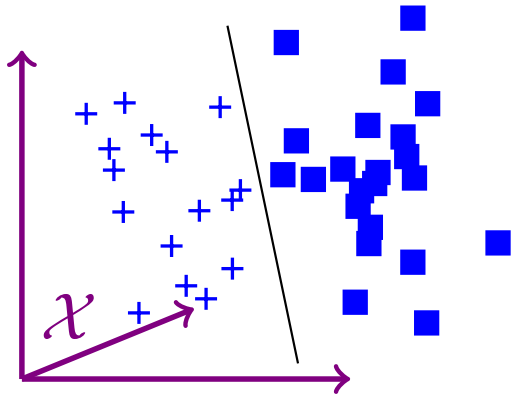


Intuition of the Optimal Transport

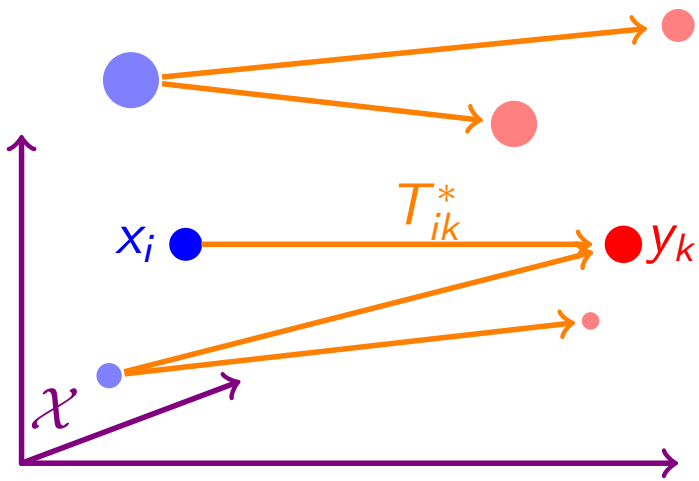


Overview

Machine Learning

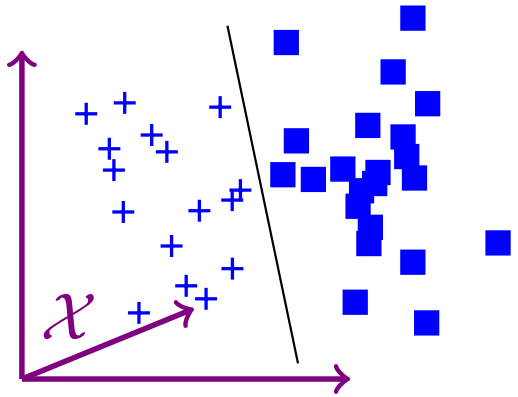


Optimal Transport

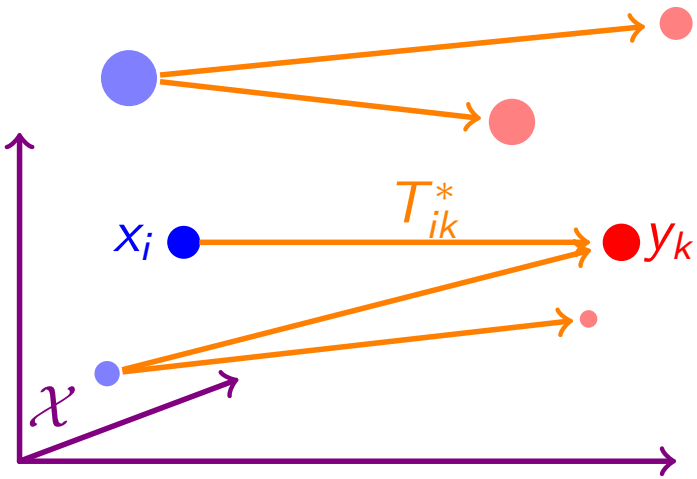


Overview

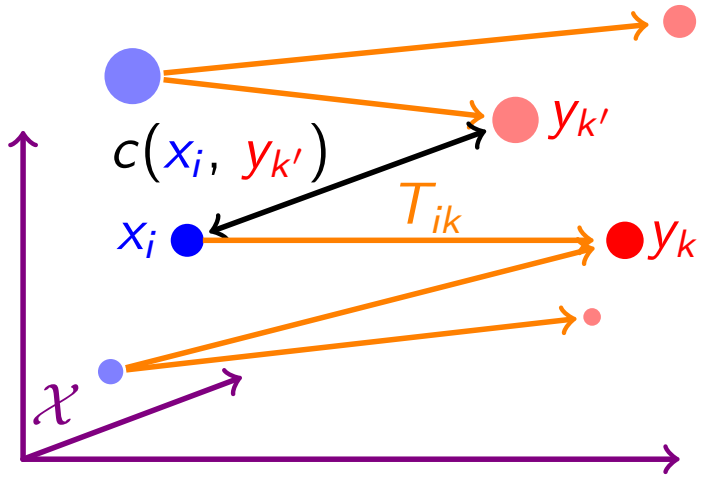
Machine Learning



Optimal Transport

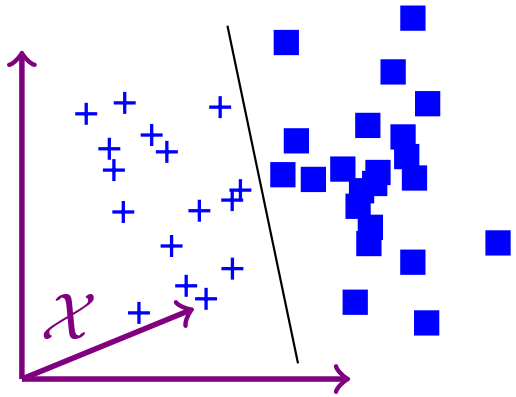


Ground Metric

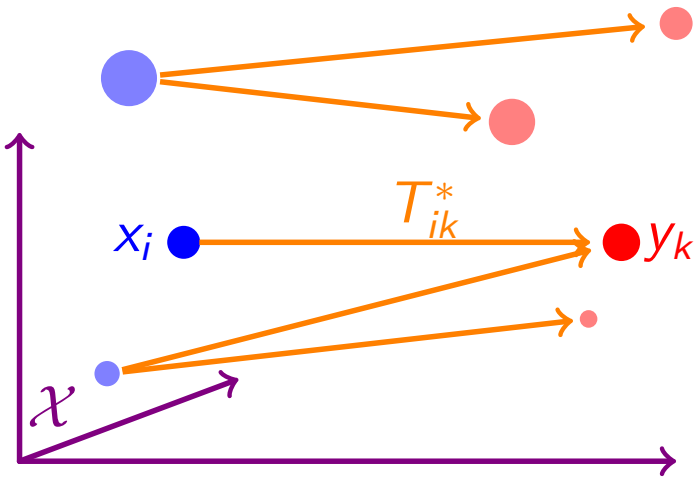


Overview

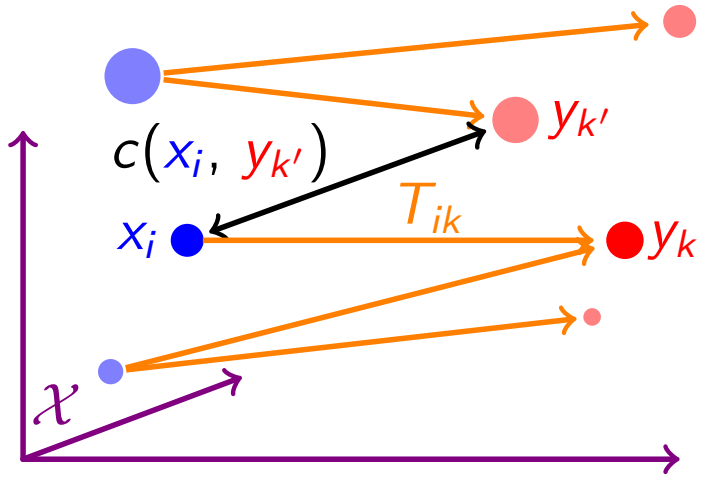
Machine Learning



Optimal Transport



Ground Metric



Generalized Framework

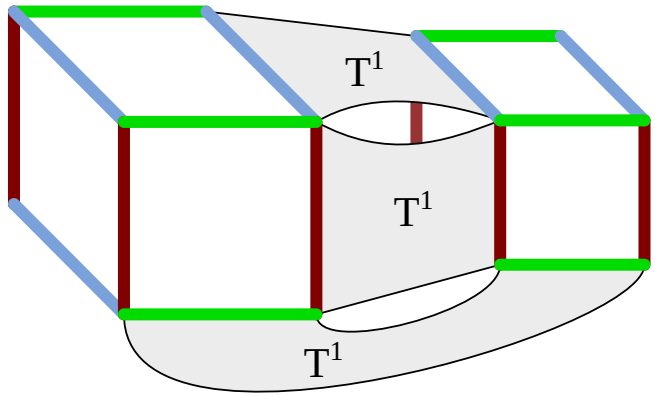


Table of Contents

Discrete Optimal Transport

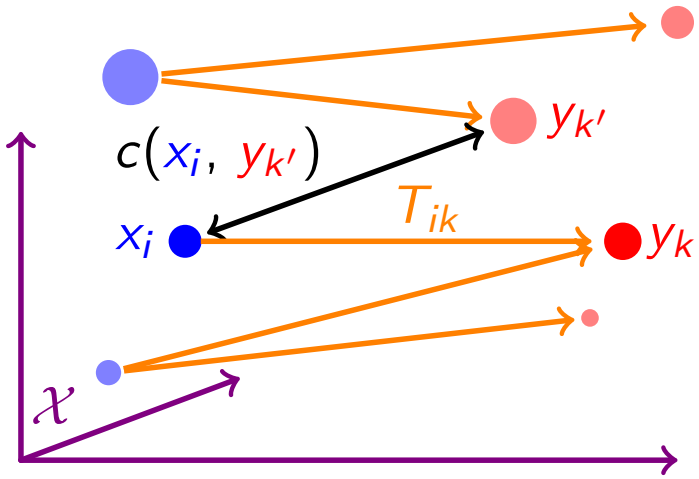


Figure 1: Optimal Transport between two distributions using the ground cost c to compare points.

Discrete Optimal Transport

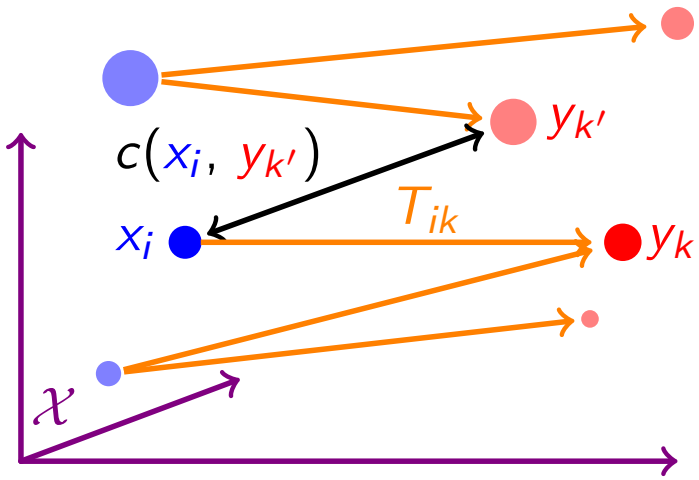


Figure 1: Optimal Transport between two distributions using the ground cost c to compare points.

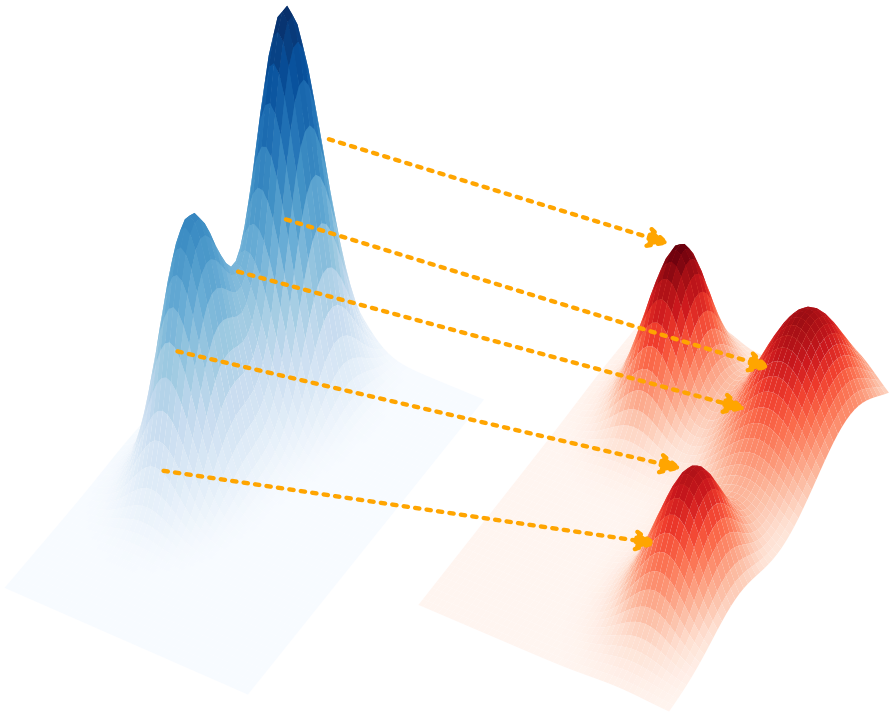
Optimal Transport [Monge, 1781; Kantorovich, 1942; Villani, 2008]

- Two empirical distributions: $\mu = \sum_{i=1}^{I} a_i \delta_{x_i}$ and $\nu = \sum_{k=1}^{K} b_k \delta_{y_k}$.
- Set of transport plans: $\Pi(\mu, \nu) = \{T \in \mathbb{R}_+^{I \times K} \,|\, T^\top \mathbb{1}_I = b, T\mathbb{1}_K = a\}$

$$\min_{T \in \Pi(\mu,\nu)} \sum_{i,k=1}^{I,K} c(x_i, y_k) T_{ik} = \min_{T \in \Pi(\mu,\nu)} \langle C, T \rangle \tag{1}$$

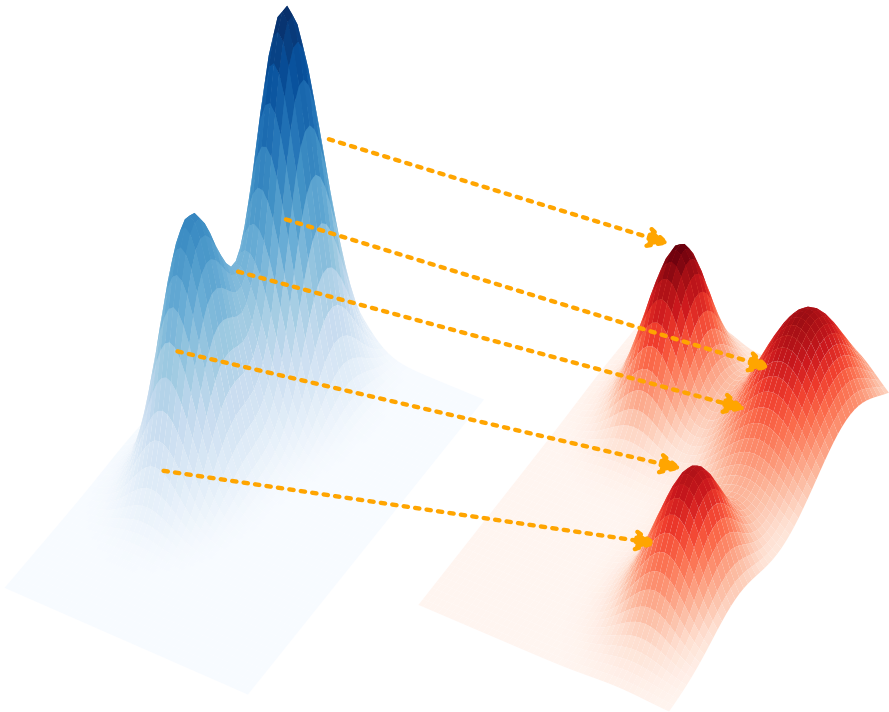
Continuous Optimal Transport (OT)

Figure 2: Optimal Transport between two continuous distributions $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$.



Continuous Optimal Transport (OT)

Figure 2: Optimal Transport between two continuous distributions $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$.



Optimal Transport [Kantorovich, 1942; Villani, 2008]

- Set of transport plans: $\Pi(\mu, \nu)$ containing all $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ with marginal μ and ν.

$$\min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\pi(x,y) \tag{2}$$

Possible solvers

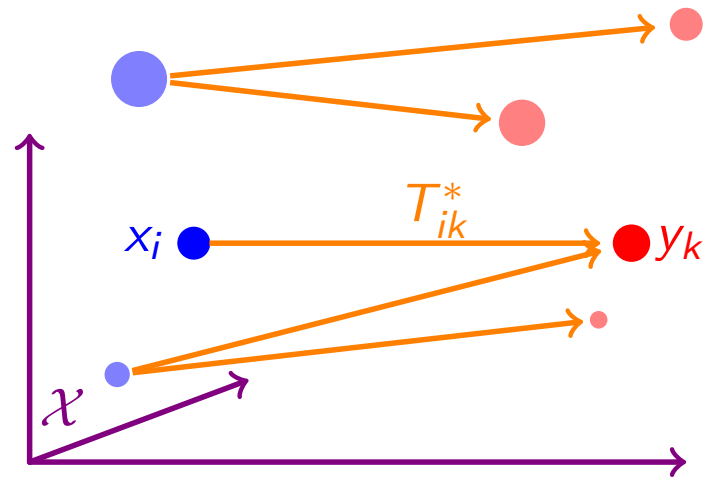
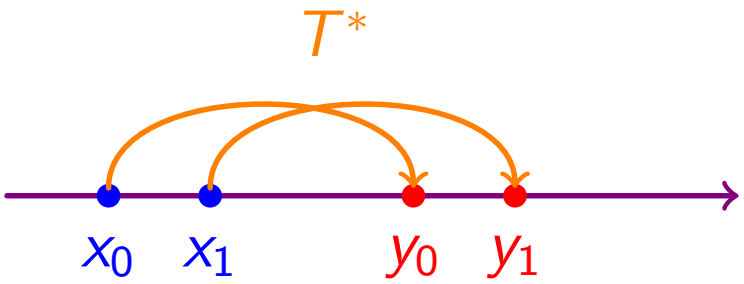


Figure 3: EMD: Based on the min cost flow problem [Ahuja et al., 1988] and $N^3 log(N)$ in the worst case.



$$c(x, y) = (x - y)^2$$

Figure 4: 1D Optimal Transport: Can be computed in $Nlog(N)$ by sorting the two lists.

Entropy regularization

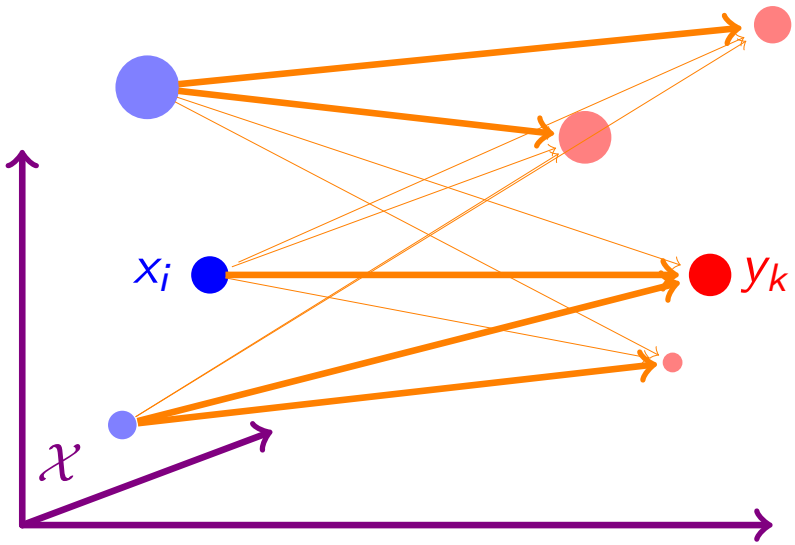


Figure 5: Solution of the entropy regularized Optimal Transport problem. The transport plan T is not sparse anymore.

Entropy regularization

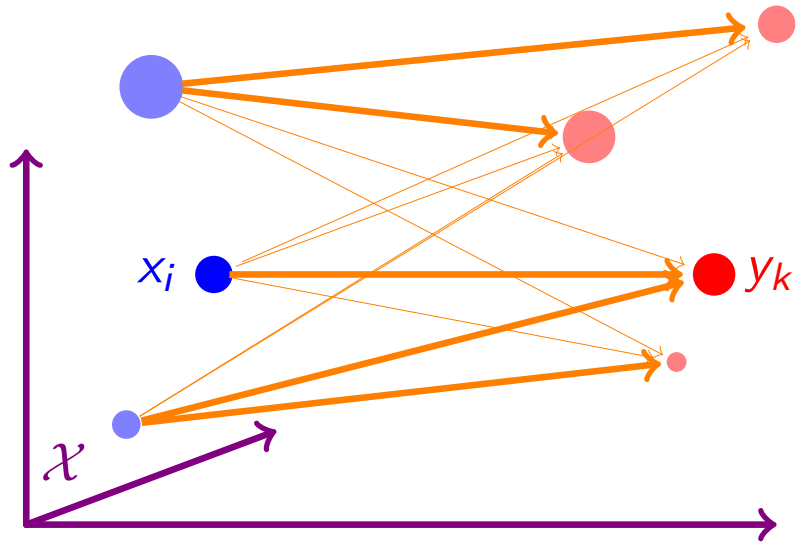


Figure 5: Solution of the entropy regularized Optimal Transport problem. The transport plan T is not sparse anymore.

Regularized OT Cuturi [2013]; Sinkhorn and Knopp [1967]

- Kullback-Leibler regularization $KL(T||T') = \sum_{i,k=1}^{I,K} T_{ik} log(\frac{T_{ik}}{T'_{ik}})$:

$$\min_{T \in \Pi(\mu,\nu)} \sum_{i,k=1}^{I,K} c(x_i, y_k) T_{ik} + \epsilon KL(T||ab^\top) \tag{3}$$

Entropy regularization

- Unique solution
- The optimal transport plan changes continuously
- Can take into account a prior on the transport plan
- Fast and simple solver available ($P \times N^2$)

Regularized OT Cuturi [2013]; Sinkhorn and Knopp [1967]

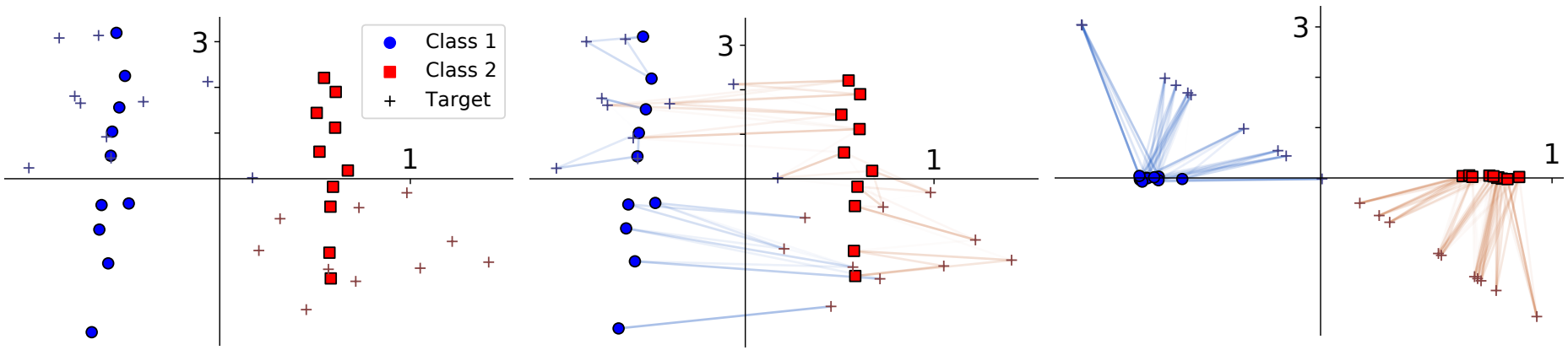
- Kullback-Leibler regularization $KL(T||T') = \sum_{i,k=1}^{I,K} T_{ik} log(\frac{T_{ik}}{T'_{ik}})$:

$$\min_{T \in \Pi(\mu,\nu)} \sum_{i,k=1}^{I,K} c(x_i, y_k) T_{ik} + \epsilon KL(T||ab^\top) \tag{3}$$

Table of Contents

Metric Learning for Optimal Transport

Based on a published paper at the International Joint Conference on Artificial Intelligence (IJCAI) [Kerdoncuff et al., 2020]



Domain Adaptation (DA): Intuition

- Internship supervision of Thibaud Leteno, in collaboration with DGFiP, on companies **fraud detection**.

1. Usual Goal: Predict the companies that might defraud
2. Goal in **DA** context: Predict the companies that might defraud in a **Covid crisis** context

Intuition of Optimal Transport for Domain Adaptation

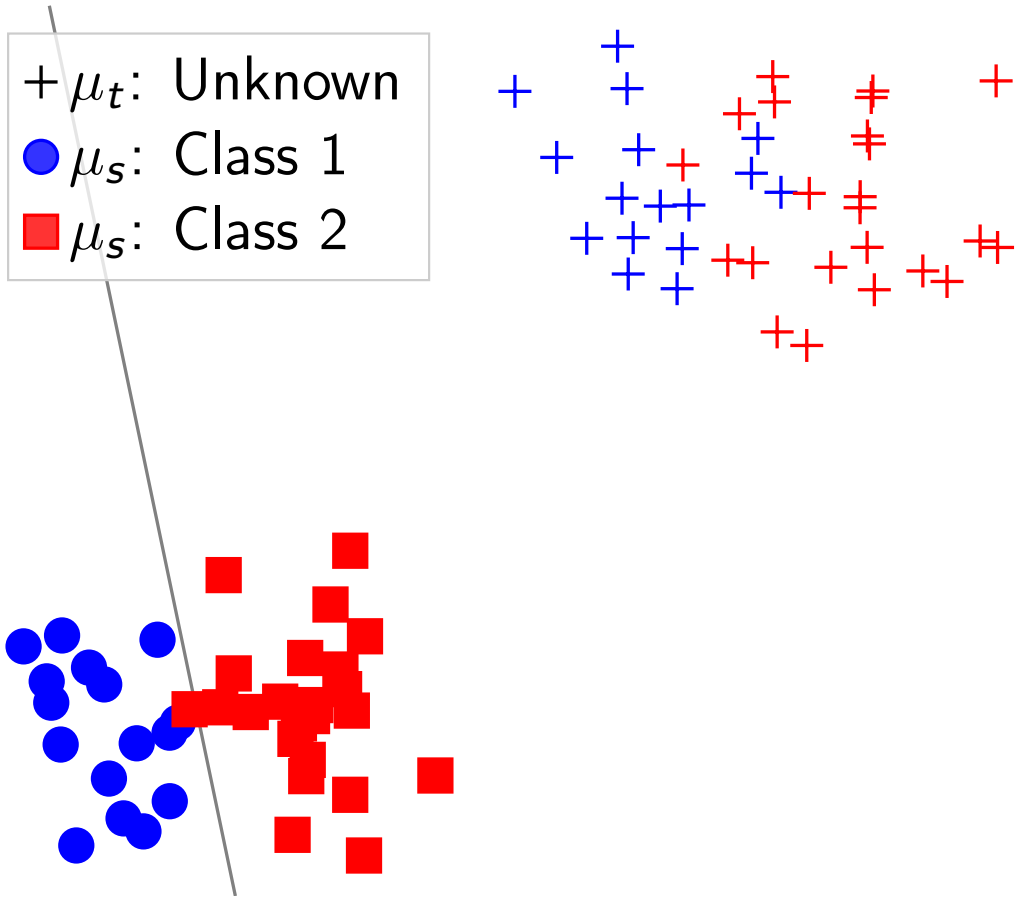


Figure 5: Optimal Transport for Domain Adaptation

Intuition of Optimal Transport for Domain Adaptation

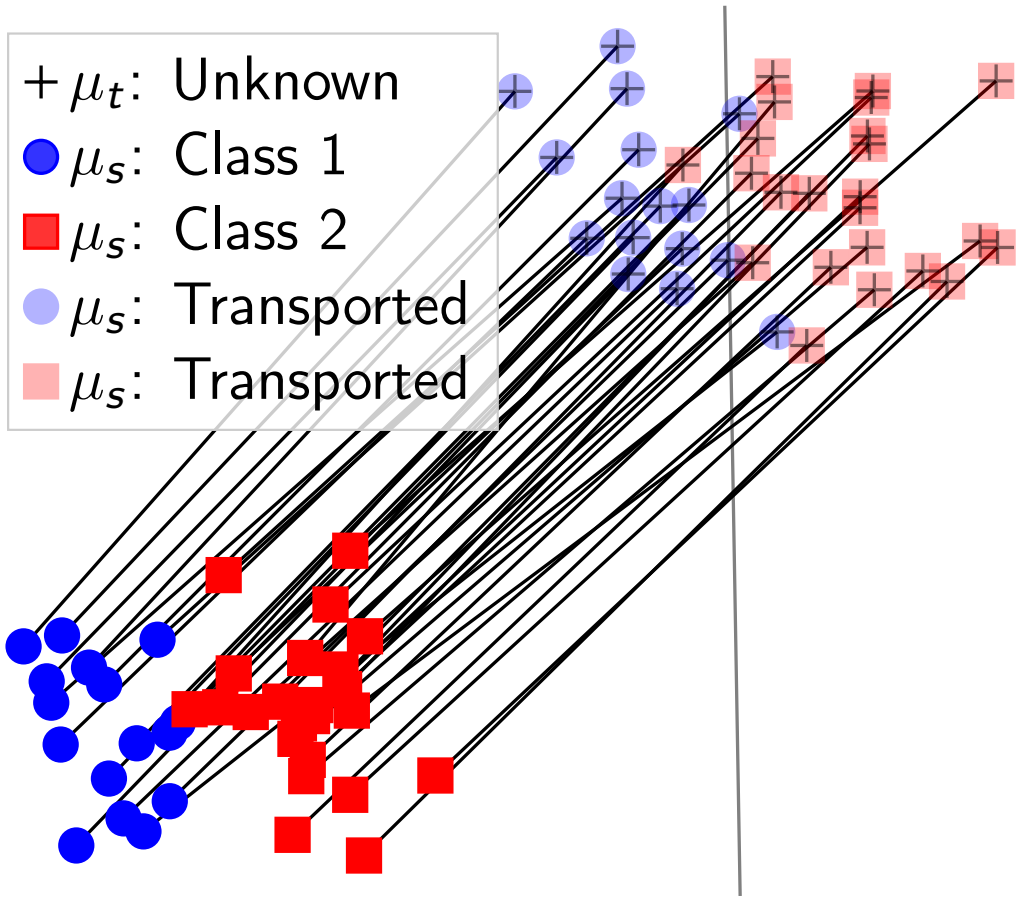


Figure 6: Optimal Transport for Domain Adaptation

Intuition of Optimal Transport for Domain Adaptation

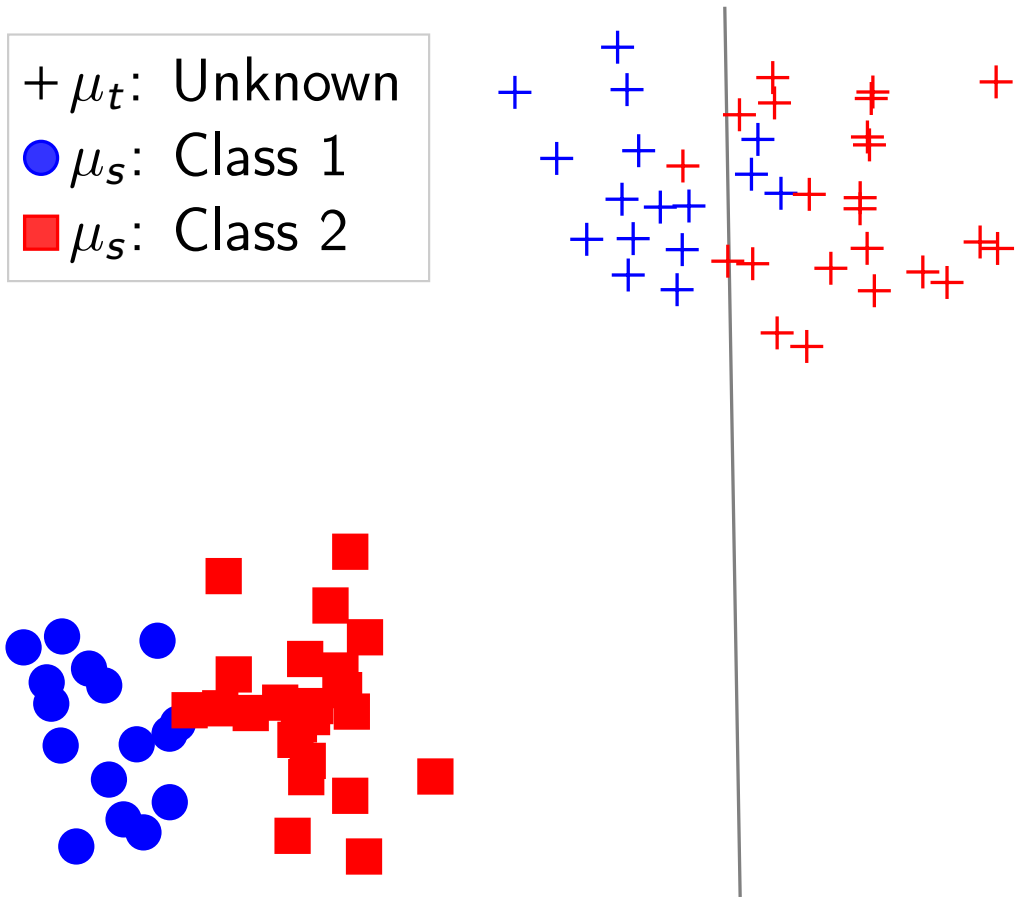


Figure 7: Optimal Transport for Domain Adaptation

Optimal Transport for Domain Adaptation

Optimal Transport for Domain Adaptation [Courty et al., 2017]

$$\min_{T \in \Pi(\hat{\mu}_s, \hat{\mu}_t)} \langle T, C \rangle + \epsilon KL(T || ab^\top) + \lambda_c \left(\sum_{k=1}^{K} \sum_{cl=1}^{\#classes} \| T(\mathcal{I}_{cl}, k) \|_2 \right) \quad (4)$$

✓ Points of different classes should **not** be sent to same location

Metric learning

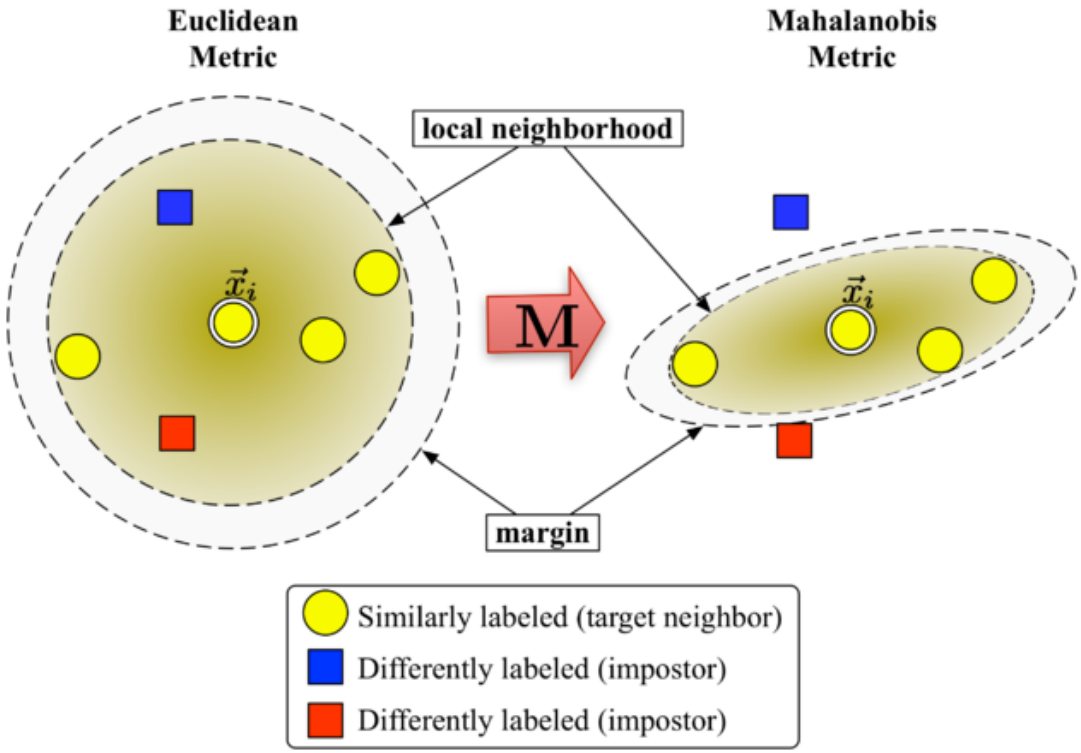


Figure 8: LMNN [Weinberger and Saul, 2009] CC-BY Mlguy (wikipedia)

Mahalanobis metric

- $C_M^2(x, x') = (x - x')M(x - x')^T = \|Lx - Lx'\|_2^2$ with $L^T L = M$

Metric Learning for Optimal Transport (MLOT)

Algorithm 1 MLOT

Require: η (gradient step) $\boldsymbol{X}^s$ $\boldsymbol{X}^t$ $\boldsymbol{Y}^s$

1: $\boldsymbol{V}_s = PCA(\boldsymbol{X}^s)$, $\boldsymbol{V}_t = PCA(\boldsymbol{X}^t)$

2: $\boldsymbol{L}_s = \boldsymbol{V}_s^\top \boldsymbol{V}_s$, $\boldsymbol{L}_t = \boldsymbol{V}_t^\top \boldsymbol{V}_t$

3: **for** $i = 1$ **to** P **do**

4: $\quad T = \underset{T \in \Pi(\hat{\mu}_s, \hat{\mu}_t)}{\operatorname{argmin}} \langle T, C^2(\boldsymbol{L}_s, \boldsymbol{L}_t) \rangle - \epsilon \mathcal{H}(T) + \lambda_c \Omega_{cl}(T)$

5: $\quad \boldsymbol{L}_s = \boldsymbol{L}_s - \eta \nabla_{\boldsymbol{L}_s} (\langle T, C^2(\boldsymbol{L}_s, \boldsymbol{L}_t) \rangle + \lambda_l \Omega_l(\boldsymbol{L}_s))$

6: **end for**

7: $\tilde{\boldsymbol{X}}_s = T \boldsymbol{L}_t \boldsymbol{X}^t$

8: classifier = classifier method$(\tilde{\boldsymbol{X}}_s, \boldsymbol{Y}^s)$

9: $\hat{\boldsymbol{Y}}^t$ = classifier$(\boldsymbol{X}^t)$

10: **return** $\hat{\boldsymbol{Y}}^t$

Illustration of the usefulness of Metric Learning

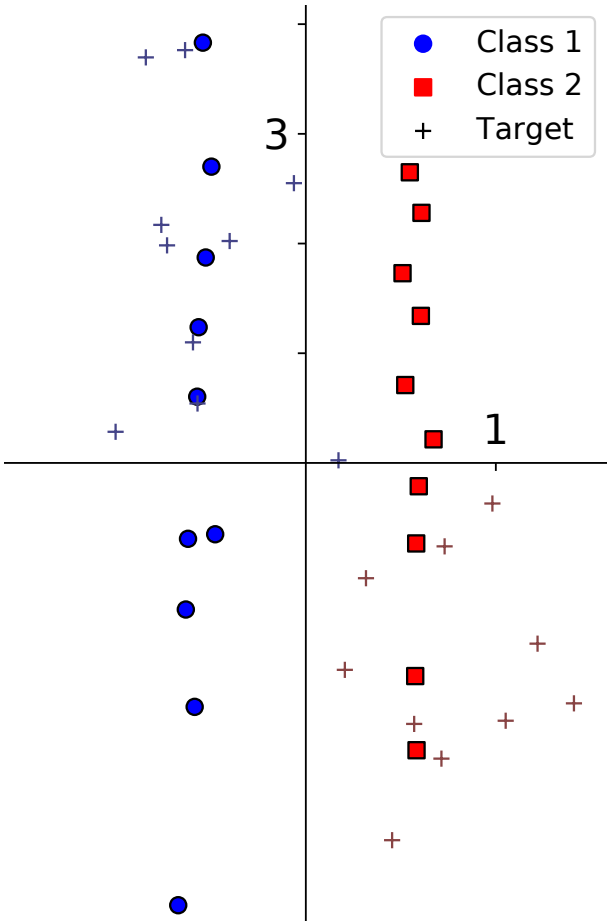


Figure 9: Dataset

Illustration of the usefulness of Metric Learning

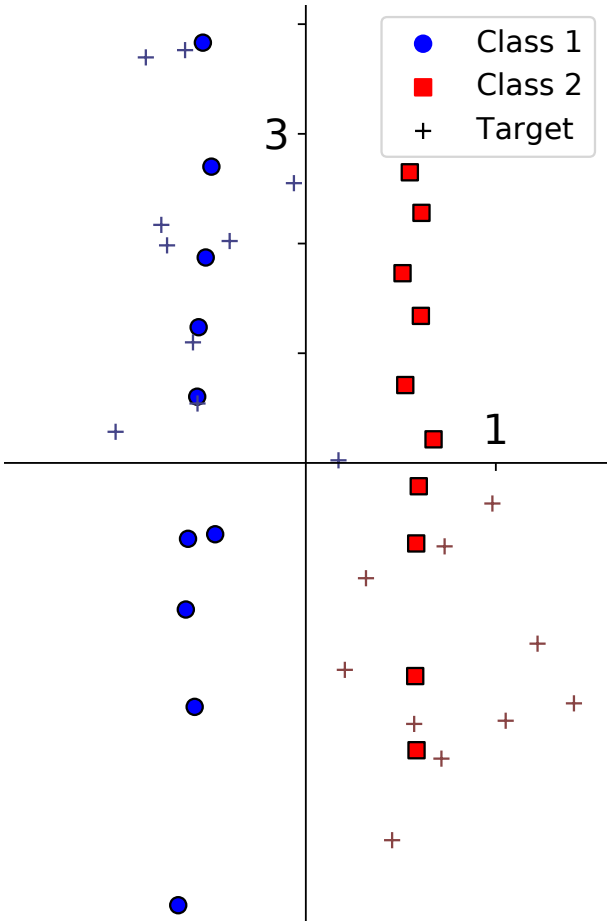


Figure 9: Dataset

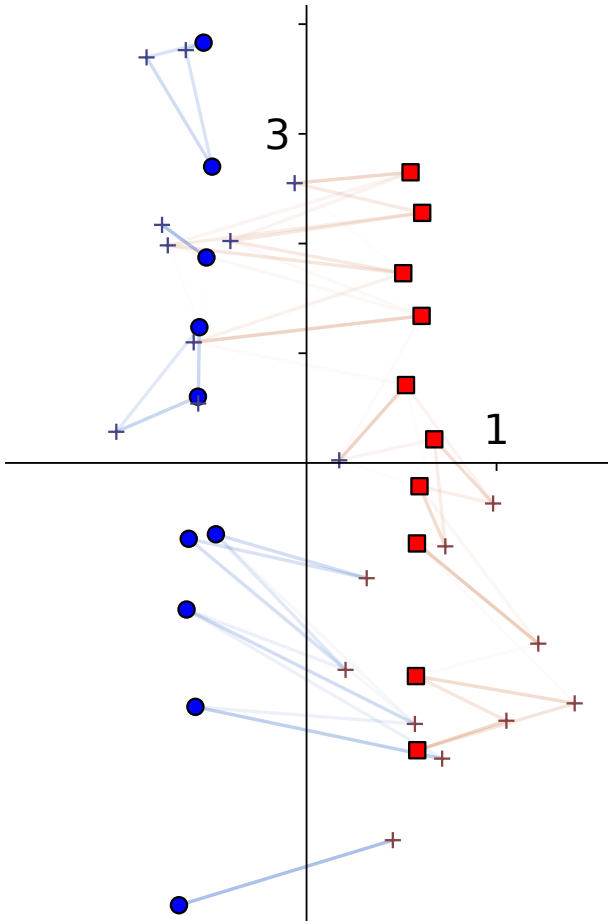


Figure 10: OTDA

Illustration of the usefulness of Metric Learning

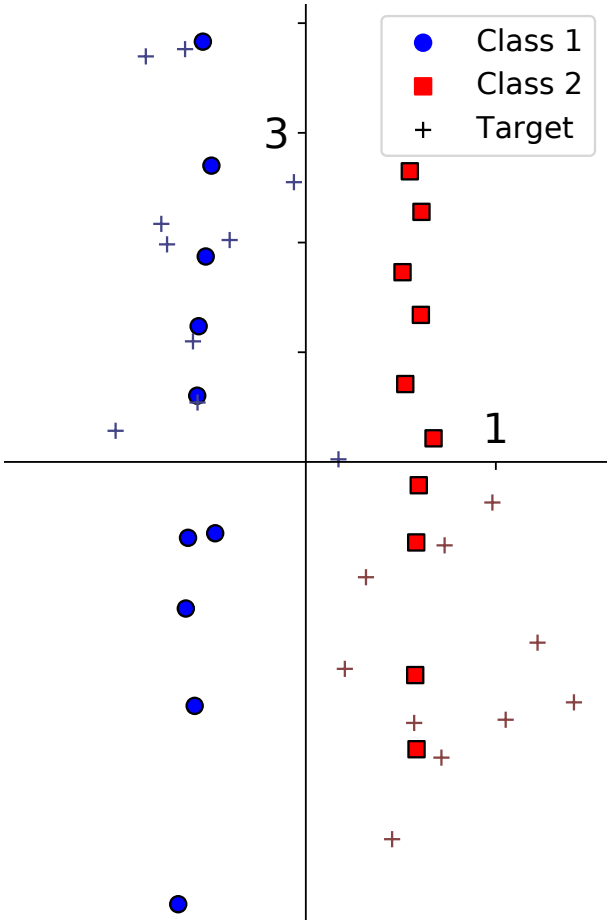


Figure 9: Dataset

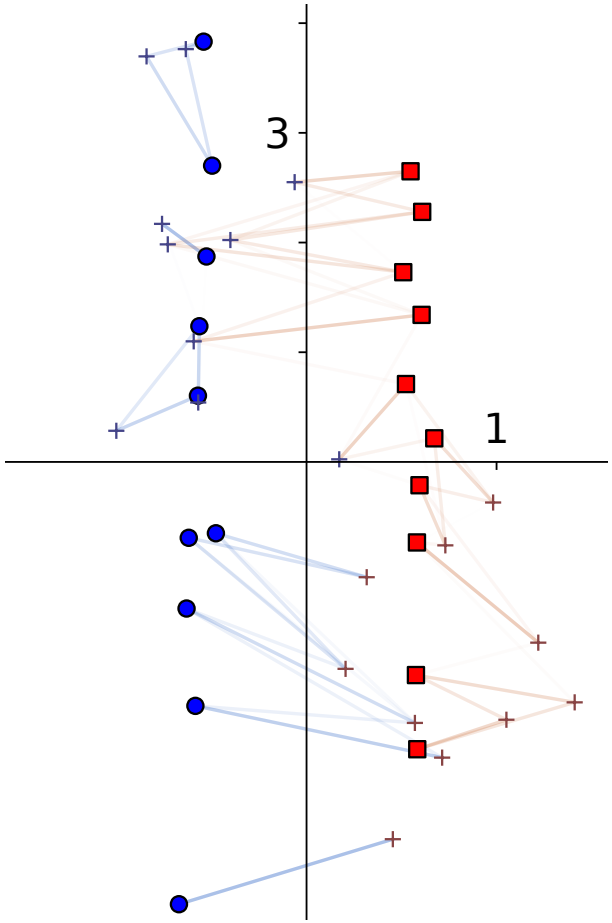


Figure 10: OTDA

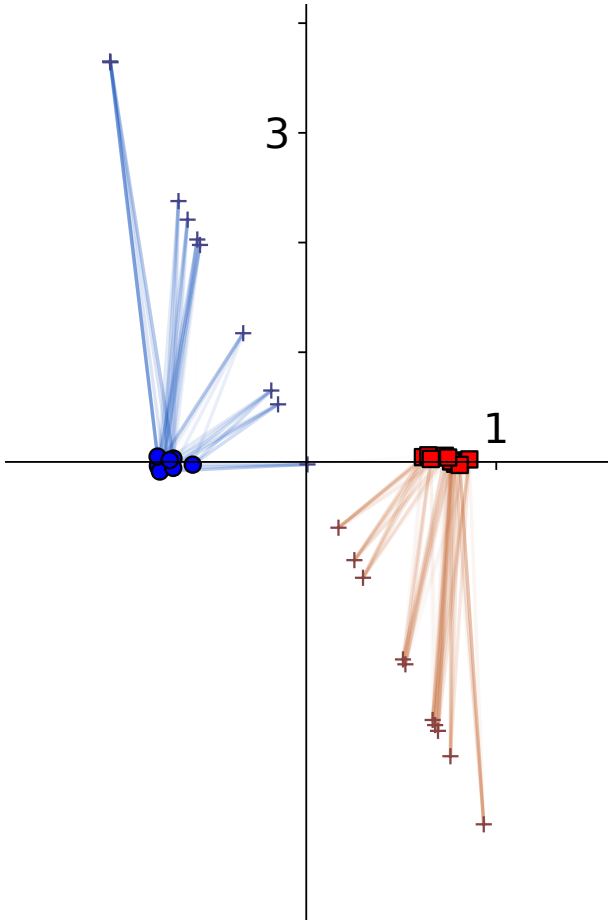


Figure 11: MLOT

Cross validation of hyperparameters

- In **Unsupervised** Domain Adaptation the target labels are **not available**
- We use the Reverse Validation of Zhong et al. [2010]

Method	OT	TCA	LMNN	SA	JDOT	OTDA	OTDA_p	MLOT
Classical Validation	45.3	45.3	47.9	47.4	48.5	52.8	54	**55.1**
Reverse Validation	42.2	44.5	44.8	45.8	47.0	48.2	48.8	**49.7**

Table 1: Accuracy on Office-Caltech dataset [Gong et al., 2012] with SURF features.

OTDA vs MLOT

Figure 12: MLOT - OTDA

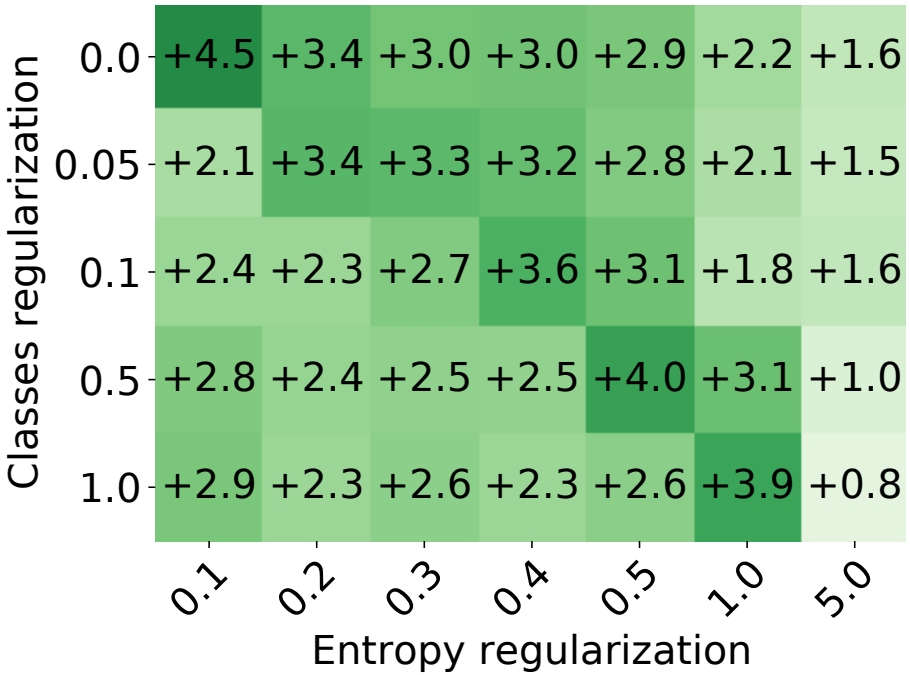


Figure 13: MLOT

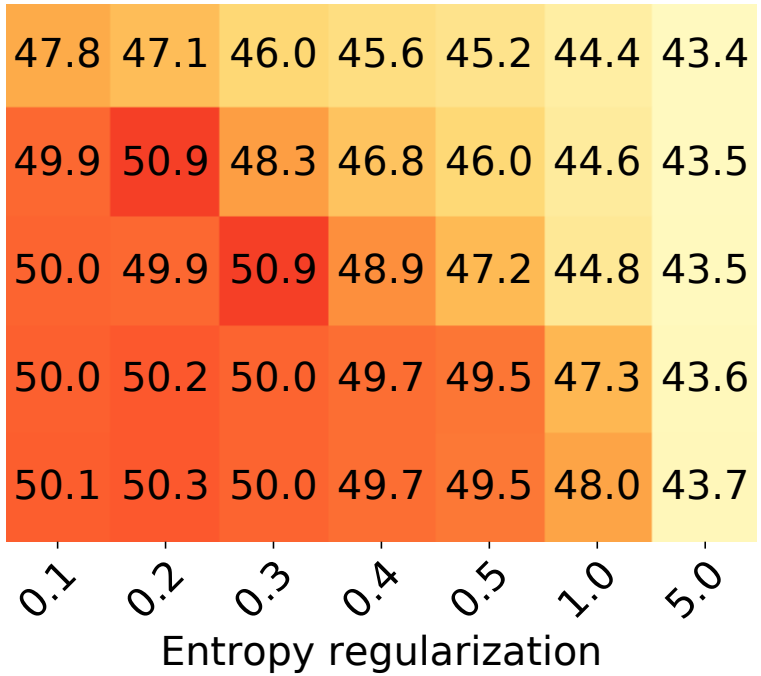
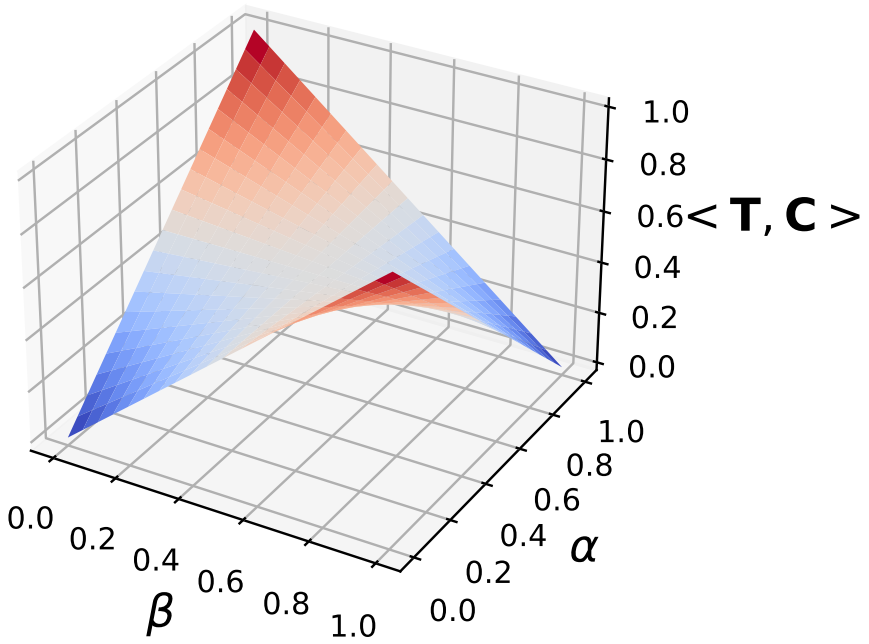


Figure 14: Accuracy of OTDA and MLOT on Office-Caltech dataset [Gong et al., 2012] with SURF features.

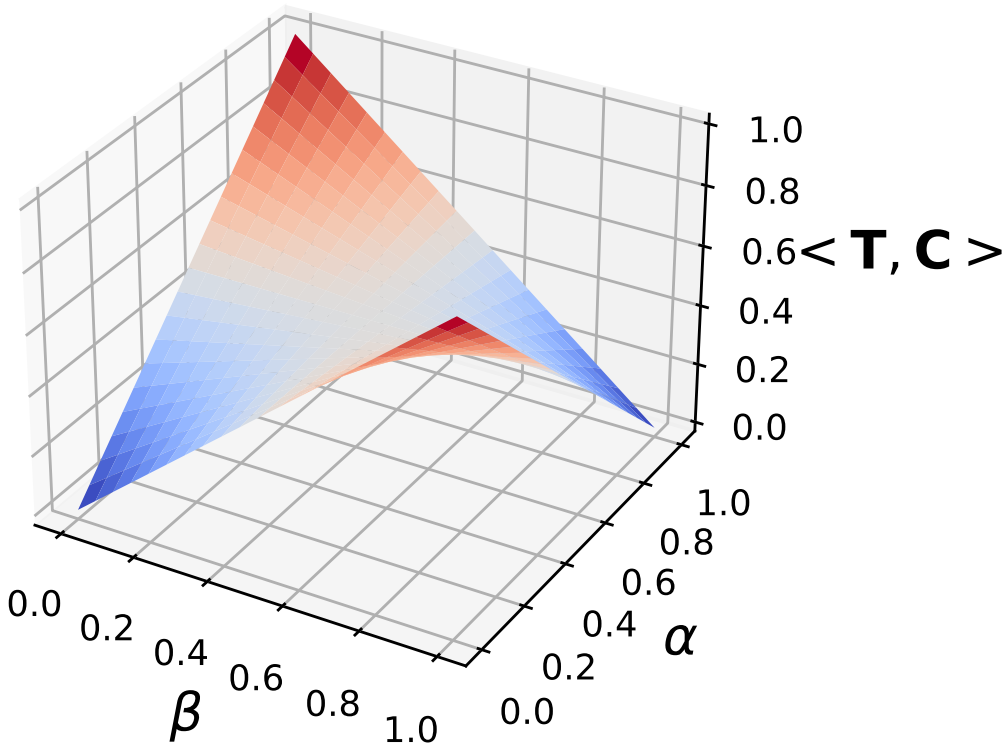
Table of Contents

A Swiss Army Knife for Minimax Optimal Transport

Based on a published paper at International Conference on Machine Learning (ICML) 2020 [Dhouib et al., 2020]



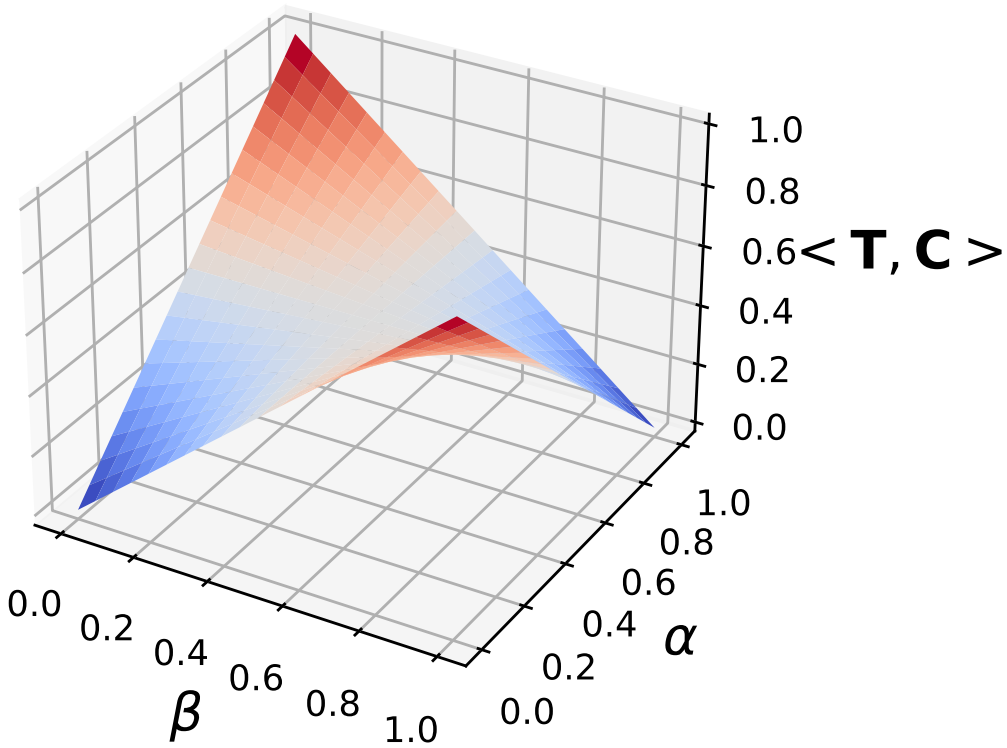
Intuition of the OT Minimax problem



Intuition of the OT Minimax problem

Cost matrices

- $C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- $C_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- $C = \alpha C_1 + (1-\alpha) C_2$



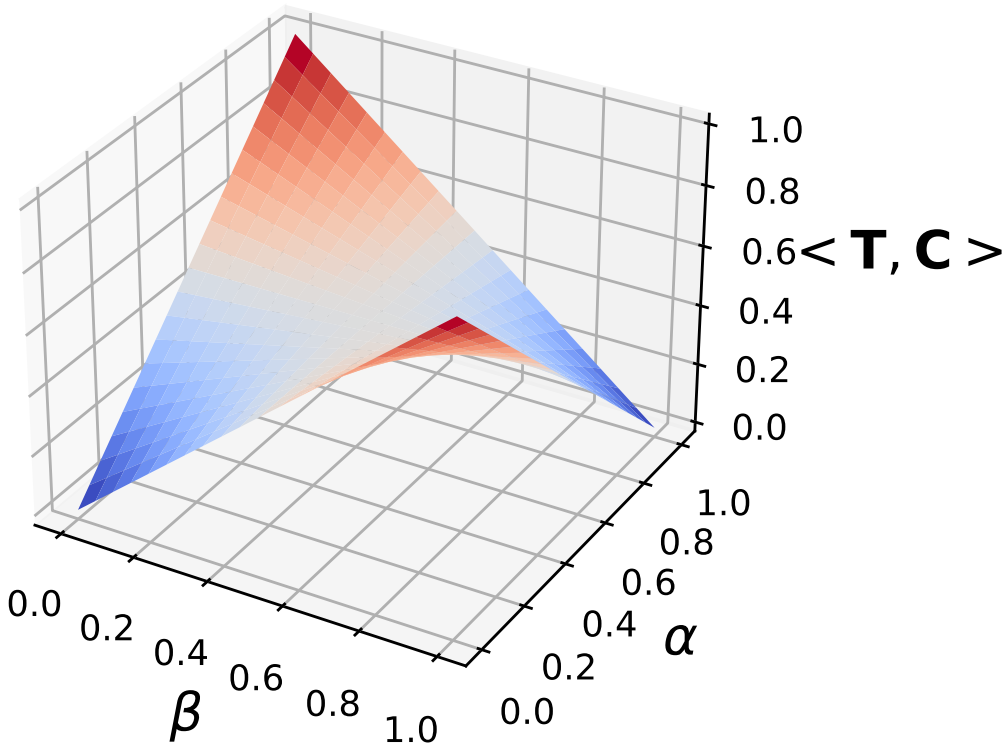
Intuition of the OT Minimax problem

Cost matrices

- $C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- $C_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- $C = \alpha C_1 + (1-\alpha) C_2$

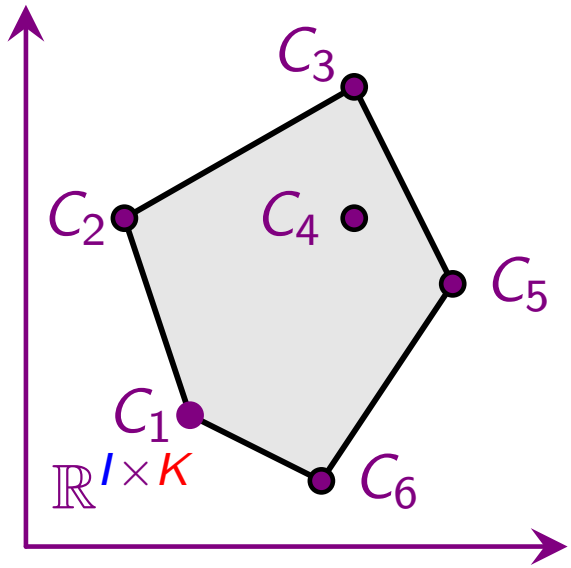
Associated Transport Plan

- $T_1^* = \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix}$
- $T_2^* = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$
- $T = \beta T_1^* + (1-\beta) T_2^*$



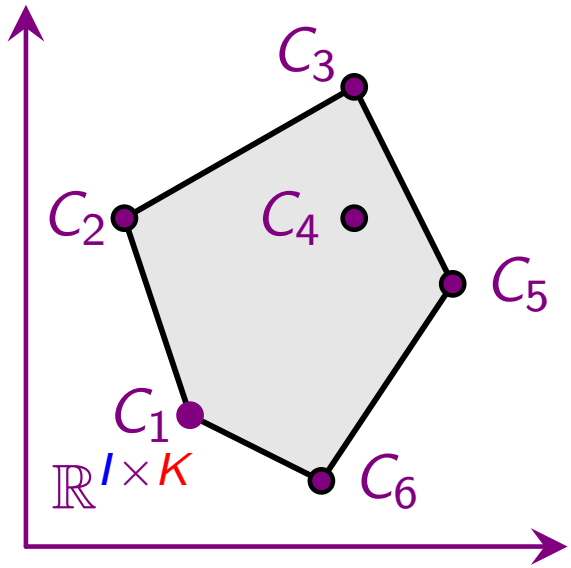
Different type of set

- Convex hull of finite set of cost matrices:

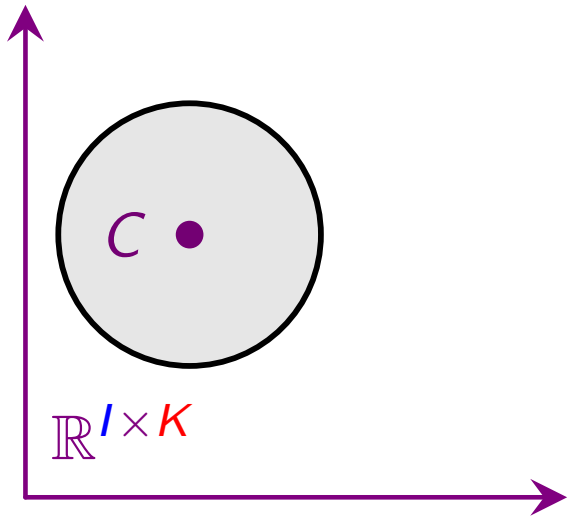


Different type of set

- Convex hull of finite set of cost matrices:



- Mahalanobis ball centered at a matrix C :



Cutting set algorithm [Mutapcic and Boyd, 2009]

Proper definition of the Minimax problem

$$\mathcal{W}_{\mathcal{C}} = \min_{T \in Conv(\mathcal{P})} \max_{C \in \mathcal{C}} \langle T, C \rangle$$

Cutting set algorithm [Mutapcic and Boyd, 2009]

Proper definition of the Minimax problem

$$\mathcal{W}_{\mathcal{C}} = \min_{T \in Conv(\mathcal{P})} \max_{C \in \mathcal{C}} \langle T, C \rangle \quad \Longleftrightarrow \quad \operatorname*{argmax}_{C \in \mathcal{C}, \omega \geq 0} \ \omega \tag{5}$$
$$\text{s.t.} \ \langle T, C \rangle \geq \ \omega, \ \forall T \in \mathcal{P}$$

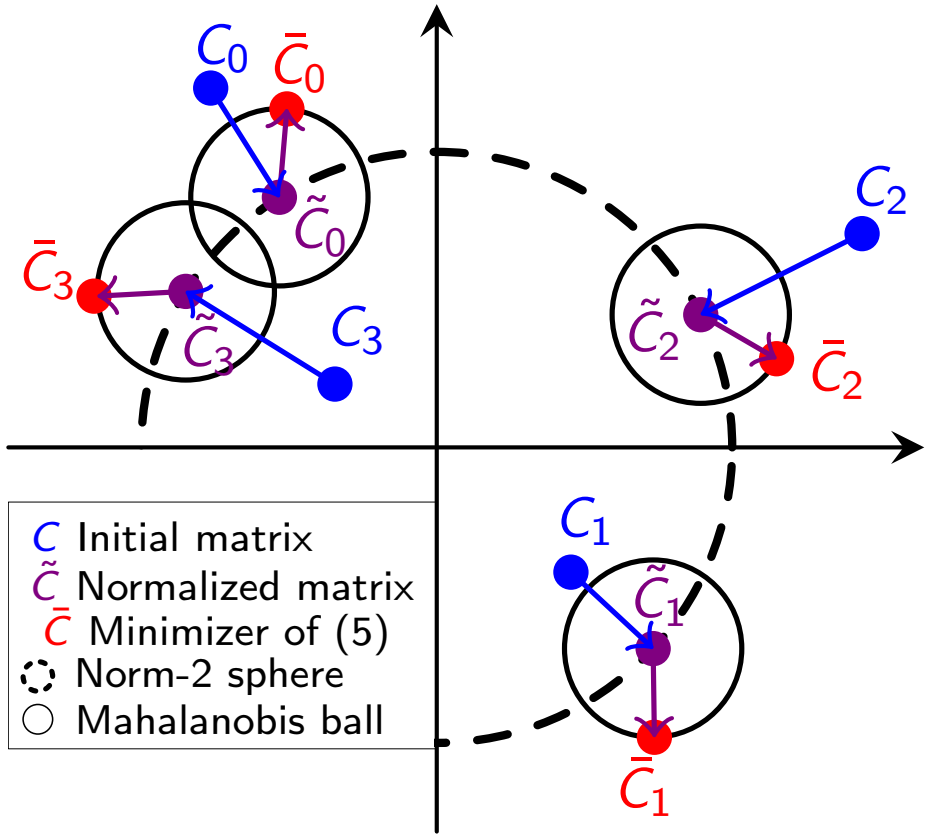
Algorithm 3 Cutting set method for RKP$(\Pi, \mathcal{C})$

1: **Input:** $\mathcal{C}$, $\mathcal{P}_0 \subset \Pi$, thd
2: **while** $err >$ thd **do**
3: Solve (5) to obtain (ω_t, C_t) and $(q_0, ..., q_{|\mathcal{P}_t|-1})$
4: Find $T_t \in \underset{T \in \Pi}{argmin} \langle T, C_t \rangle$
5: $err \leftarrow (\omega_t - \langle T_t, C_t \rangle) / \langle T_t, C_t \rangle$
6: $\mathcal{P}_{t+1} = \mathcal{P}_t \cup \{T_t\}$
7: $t \leftarrow t + 1$
8: **end while**
9: **return** $\sum_{l=0}^{|\mathcal{P}_t|-1} q_l T_l, \ C_t$

Wasserstein stability

Wasserstein stability of a matrix C induce by μ and ν

$$\mathcal{WS}_C = \mathcal{W}_{\mathcal{C}_C}(\mu, \nu) - \mathcal{W}_C(\mu, \nu)$$



Usefulness of the Wasserstein stability (Color transfer)

Usefulness of the Wasserstein stability (Color transfer)

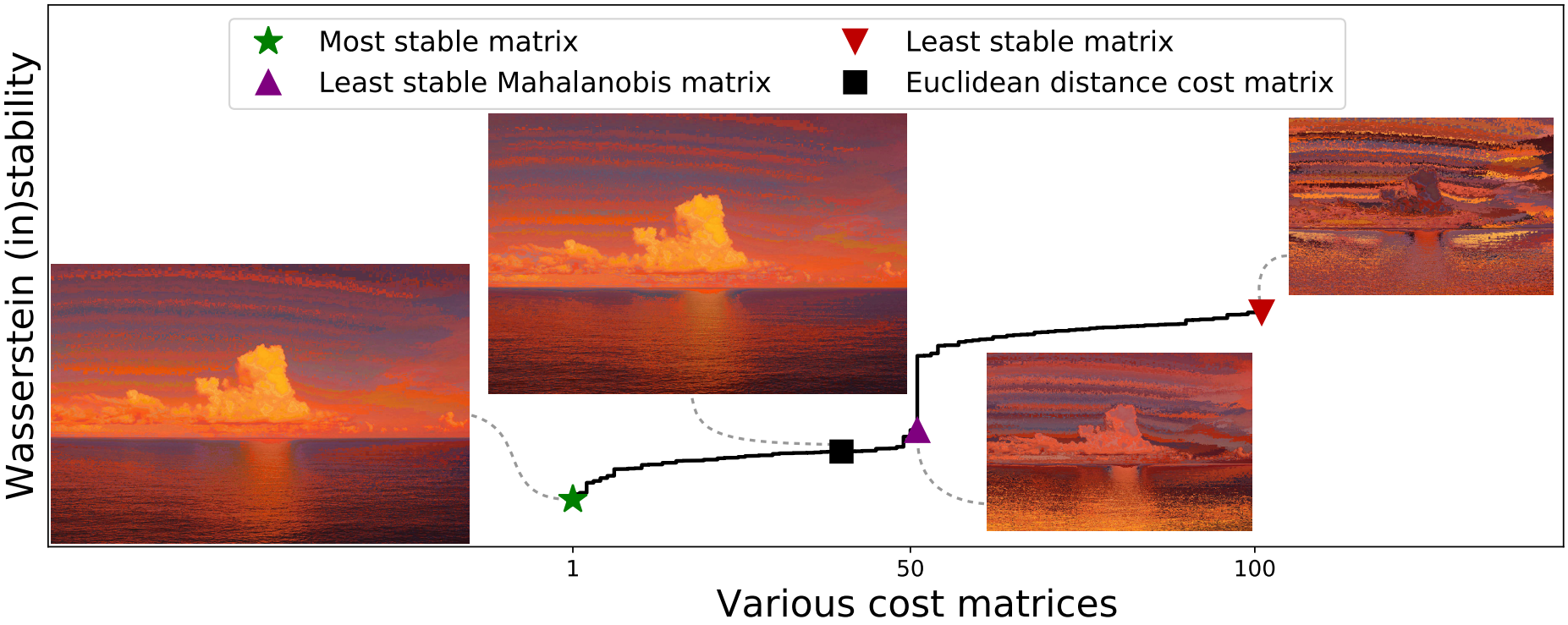
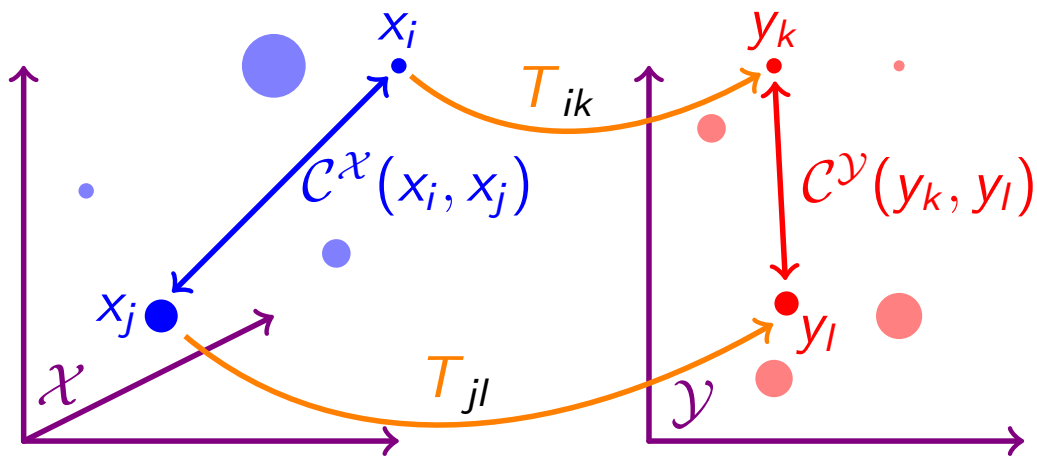


Table of Contents

Sampled Gromov Wasserstein

Based on a published paper in the Machine Learning Journal (MLJ) [Kerdoncuff et al., 2021] and presented at the ECML-PKDD 2021 conference.



The Gromov Wasserstein problem

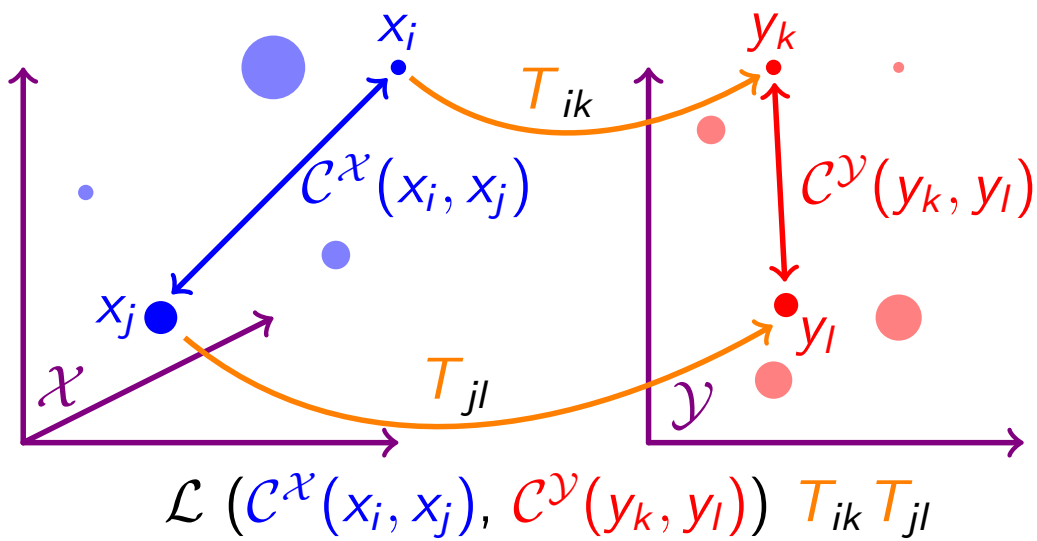


Figure 15: Gromov Wasserstein between two distributions in two vector spaces $\mathcal{X}$ and $\mathcal{Y}$, with one term from Equation (6).

Gromov Wasserstein (GW) [Memoli, 2007; Peyré et al., 2016]

- The GW distance rely only on intra-pairwise distance,

$$\min_{T \in \Pi(\mu,\nu)} \mathcal{E}(T, T) = \min_{T \in \Pi(\mu,\nu)} \sum_{ijkl=1}^{I,I,K,K} \mathcal{L}\left(\mathcal{C}^{\mathcal{X}}(x_i, x_j), \mathcal{C}^{\mathcal{Y}}(y_k, y_l)\right) T_{ik} T_{jl} \quad (6)$$

The mostly used solver: Entropy Gromov Wasserstein

Algorithm 4 Entropy Gromov Wasserstein (EGW) [Peyré et al., 2016]

Require: $a, b, \mathcal{C}^{\mathcal{X}}, \mathcal{C}^{\mathcal{Y}}, \mathcal{L}, \epsilon$

$T_0 = ab^\top$

for s= 0 **to** S-1 **do**

$\Lambda = \nabla_T \mathcal{E}(T, T) = \sum_{jl=1}^{I,K} \mathcal{L}\left(\mathcal{C}^{\mathcal{X}}(\cdot, x_j), \mathcal{C}^{\mathcal{Y}}(\cdot, y_l)\right) T_{jl}$

$T'_s = \min_{T' \in \Pi(\mu,\nu)} \langle \Lambda, T' \rangle + \epsilon KL(T' || ab^\top)$

$T_{s+1} = (1-1)T_s + 1\,T'_s$

end for

The mostly used solver: Entropy Gromov Wasserstein

Algorithm 5 Entropy Gromov Wasserstein (EGW) [Peyré et al., 2016]

Require: $a, b, \mathcal{C}^{\mathcal{X}}, \mathcal{C}^{\mathcal{Y}}, \mathcal{L}, \epsilon$

$T_0 = ab^\top$

for s= 0 **to** S-1 **do**

$\Lambda = \nabla_T \mathcal{E}(T, T) = \sum_{jl=1}^{I,K} \mathcal{L}\left(\mathcal{C}^{\mathcal{X}}(\cdot, x_j), \mathcal{C}^{\mathcal{Y}}(\cdot, y_l)\right) T_{jl}$

$T'_s = \min_{T' \in \Pi(\mu,\nu)} \langle \Lambda, T' \rangle + \epsilon KL(T' || ab^\top)$

$T_{s+1} = (1-1)T_s + 1\,T'_s$

end for

Relation with the Frank-Wolfe (FW) algorithm [Frank et al., 1956]

- EGW with no entropy ($\epsilon = 0$) $\iff$ FW with full step size ($\alpha = 1$).

Motivations and contributions

Existing limitations

- Time complexity for computing the gradient $\nabla_T \mathcal{E}(T, T)$ is $O\left(N^4\right)$
- Time complexity can be reduce to $O\left(N^3\right)$ but only for specific loss functions $\mathcal{L}(x, y) = f_1(x) + h_1(x)h_2(y) + f_2(y)$
- The existing convergence proof of EGW are valid only for high value of ϵ.

Contributions

- ✓ Approximate the gradient $\nabla_T \mathcal{E}(T, T)$ in $O\left(M \times N^2\right)$
- ✓ Propose a convergence bound to stationary points
- ✓ Explore a very fast variant with $M = 1$ using the 1D OT solver

Sampled Gromov Wasserstein idea (SaGroW)

$$\min_{T' \in \Pi(\mu,\nu)} \langle \nabla_T \mathcal{E}(T, T), T' \rangle = \min_{T' \in \Pi(\mu,\nu)} \left\langle \sum_{j,l} T_{jl} \boldsymbol{L}_{.j.l}, T' \right\rangle$$

Sampled Gromov Wasserstein idea (SaGroW)

$$\min_{T' \in \Pi(\mu,\nu)} \langle \nabla_T \mathcal{E}(T, T), T' \rangle = \min_{T' \in \Pi(\mu,\nu)} \left\langle \sum_{j,l} T_{jl} \boldsymbol{L}_{.j.l}, T' \right\rangle$$

$$= \min_{T' \in \Pi_{\mu\nu}} \langle \mathbb{E}(\Lambda), T' \rangle$$

Sampled Gromov Wasserstein idea (SaGroW)

$$\min_{T' \in \Pi(\mu,\nu)} \langle \nabla_T \mathcal{E}(T, T), T' \rangle = \min_{T' \in \Pi(\mu,\nu)} \left\langle \sum_{j,l} T_{jl} \boldsymbol{L}_{.j.l}, T' \right\rangle$$

$$= \min_{T' \in \Pi_{\mu\nu}} \langle \mathbb{E}(\Lambda), T' \rangle$$

$$\approx \min_{T' \in \Pi_{\mu\nu}} \left\langle \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{C}^m, T' \right\rangle$$

Summary of SaGroW algorithm.

Algorithm 6 SaGroW

Require: $a, b, \mathcal{C}^{\mathcal{X}}, \mathcal{C}^{\mathcal{Y}}, \mathcal{L}, M, \epsilon, \alpha$

$T_0 = ab^\top$

for s= 0 **to** S-1 **do**

  $(j_m, l_m) \sim \text{Sample}(T_s)\ \forall m \in [\![1, M]\!]$

  $\widehat{\mathbf{\Lambda}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}\left(\mathcal{C}^{\mathcal{X}}(\cdot, x_{j_m}), \mathcal{C}^{\mathcal{Y}}(\cdot, y_{l_m})\right)$

  $T'_s = \min_{T' \in \Pi(\mu,\nu)} \left\langle \widehat{\mathbf{\Lambda}}, T' \right\rangle + \epsilon KL(T' || ab^\top)$

  $T_{s+1} = (1-\alpha) T_s + \alpha T'_s$

end for

Bound for convergence to a stationary point

- The FW gap: $G(T) = \mathcal{E}(T, T) - \min_{T' \in \Pi(\mu,\nu)} \mathcal{E}(T, T')$.
- $G(T) = 0 \iff T$ is a stationary point of GW.

Bound for convergence to a stationary point

- The FW gap: $G(T) = \mathcal{E}(T, T) - \min_{T' \in \Pi(\mu,\nu)} \mathcal{E}(T, T')$.
- $G(T) = 0 \iff T$ is a stationary point of GW.

Theorem (Based on Reddi et al. [2016])

For any $\boldsymbol{L}_{ijkl} \in [0, B]$, for any distributions μ and ν with uniform weights a and b respectively, for any optimal solution T^* of Problem (6), on average for the transport plan $\overline{T}$ uniformly sampled from $(T_s)_{s \in [\![0, S-1]\!]}$, on average over all the samplings, the following bound holds:

$$\mathbb{E}\left(G\left(\overline{T}\right)\right) \leq \sqrt{\frac{2B\left(\mathcal{E}(T_0) - \mathcal{E}(T^*)\right)N}{S}} + B\sqrt{\frac{2N}{M}} + \epsilon \log(N).$$

Mirror Descent

Kullback Leiber regularization with previous iteration [Xu et al., 2019]

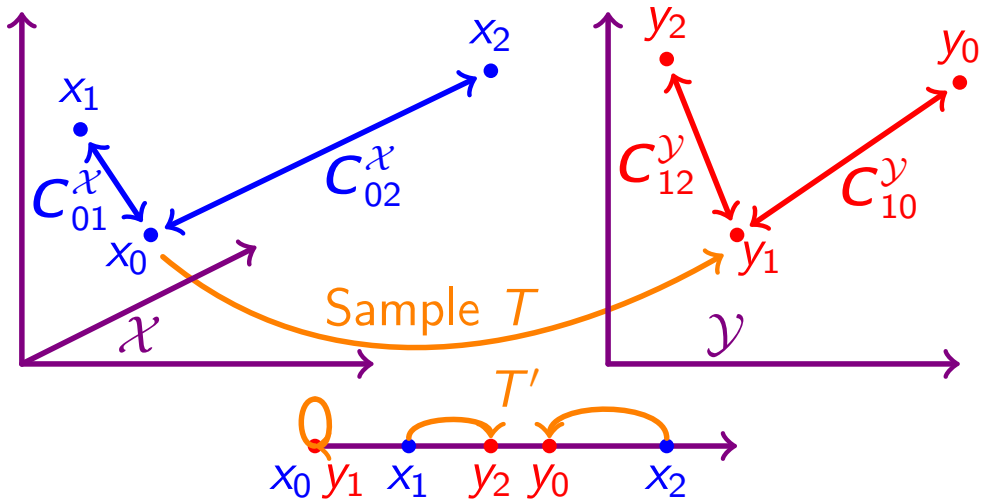
$$\left\langle \widehat{\mathbf{\Lambda}}, T' \right\rangle + \epsilon KL(T'||T) = \left\langle \widehat{\mathbf{\Lambda}} - \epsilon log(T), T' \right\rangle + \epsilon \sum_{ik} T'_{ik} log(T'_{ik}) \quad (7)$$

Algorithm 7 SaGroW with KL regularization

Require: $a, b, \mathcal{C}^{\mathcal{X}}, \mathcal{C}^{\mathcal{Y}}, \mathcal{L}, M, \epsilon, \alpha$

1: $T_0 = ab^\top$
2: **for** s= 0 **to** S-1 **do**
3: $(j_m, l_m) \sim \text{Sample}(T_s)\ \forall m \in [\![1, M]\!]$
4: $\widehat{\mathbf{\Lambda}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}\left(\mathcal{C}^{\mathcal{X}}(\cdot, x_{j_m}), \mathcal{C}^{\mathcal{Y}}(\cdot, y_{l_m})\right)$
5: $T_{s+1} = \min_{T' \in \Pi(\mu,\nu)} \left\langle \widehat{\mathbf{\Lambda}} - \epsilon log(T_s), T' \right\rangle - \epsilon \mathcal{H}(T')$
6: **end for**

Pointwise Gromov Wasserstein (PoGroW): sample $M = 1$



Speed and performances (SaGrow and PoGroW are implemented in POT)

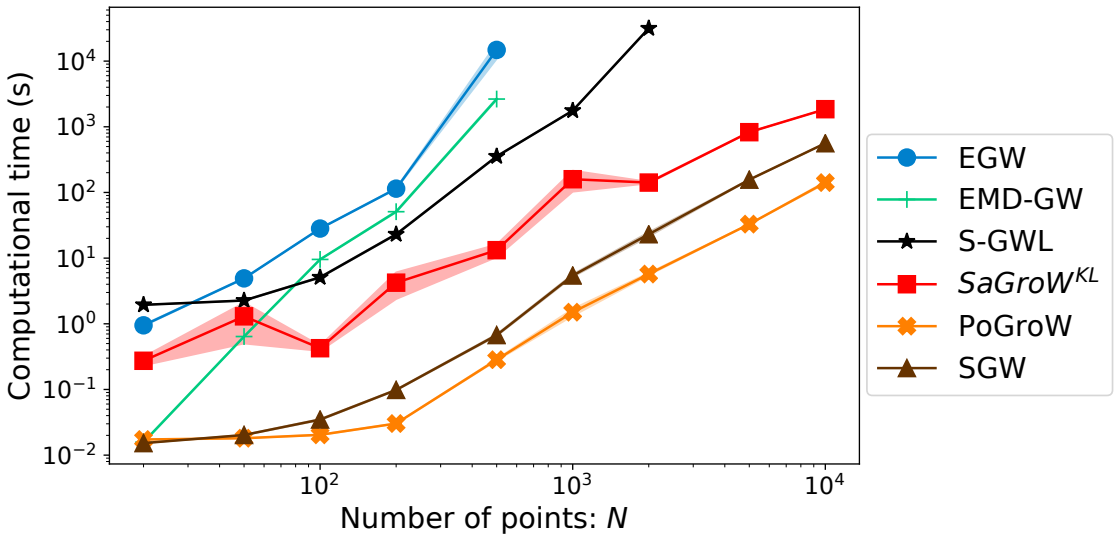


Figure 16: Computational time of various methods to compute the distance between samples from two mixtures of gaussians. The mean and the standard deviation over 10 runs are reported.

Speed and performances (SaGrow and PoGroW are implemented in POT)

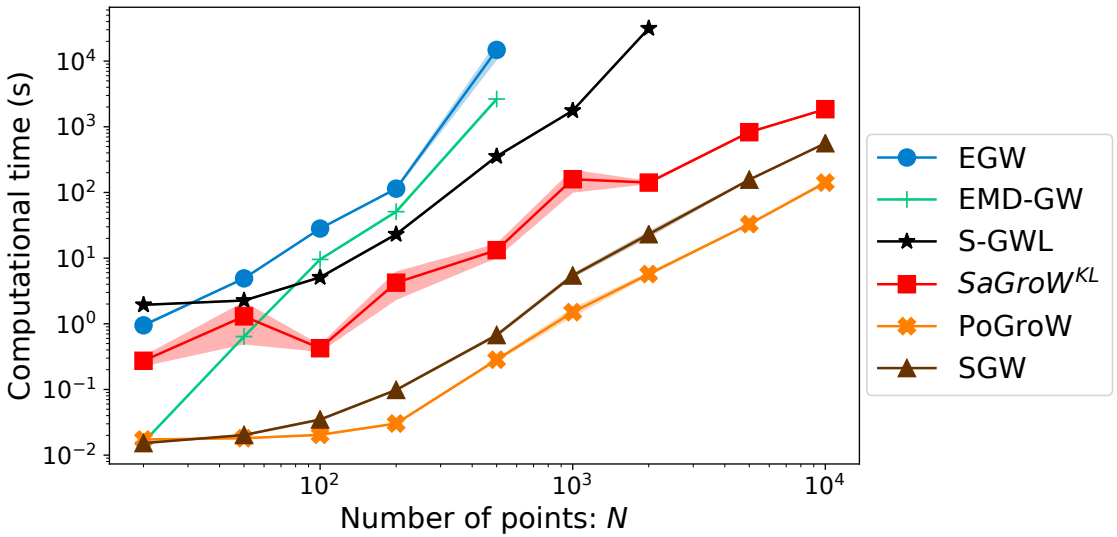


Figure 16: Computational time of various methods to compute the distance between samples from two mixtures of gaussians. The mean and the standard deviation over 10 runs are reported.

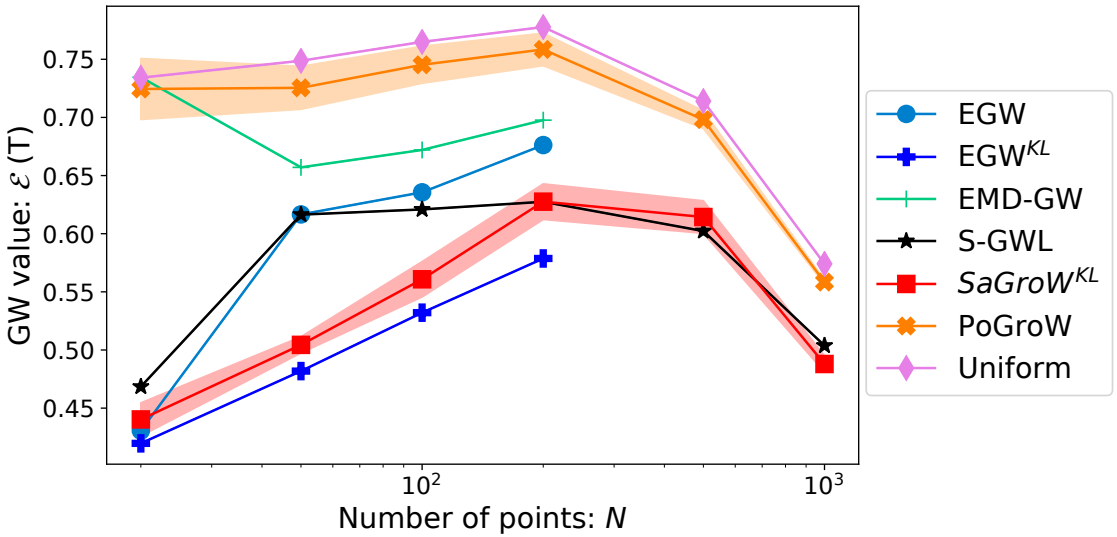
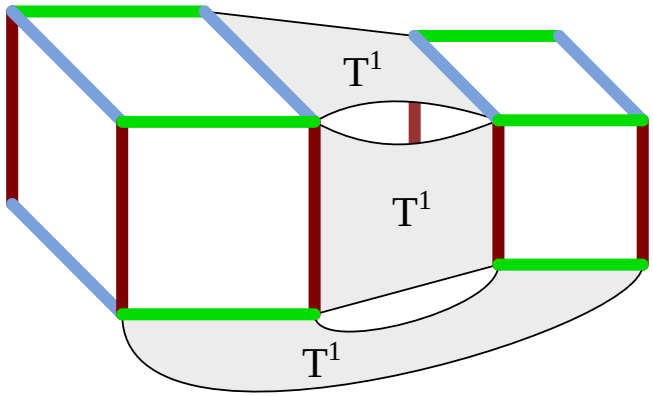


Figure 17: W distance estimation between synthetic graphs Brandes et al. [2003]. The mean and standard deviation over 10 runs are reported for the stochastic methods.

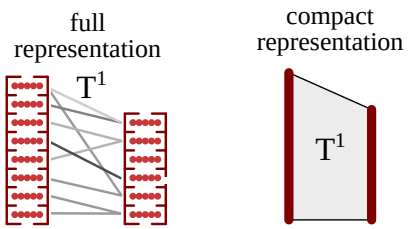
Table of Contents

Optimal Tensor Transport

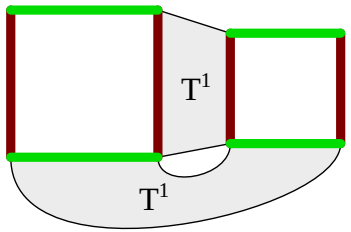
Based on an unpublished paper under submission at the AAAI conference.



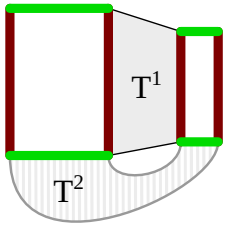
Motivation of Optimal Tensor Transport



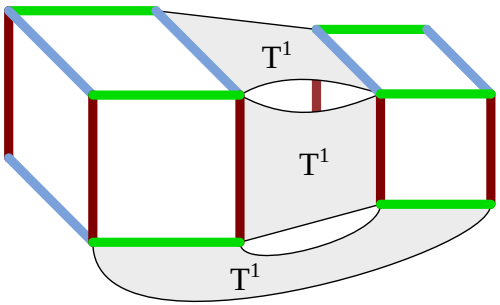
a) **OTT$_1$** (OT) *(F=5 features)*



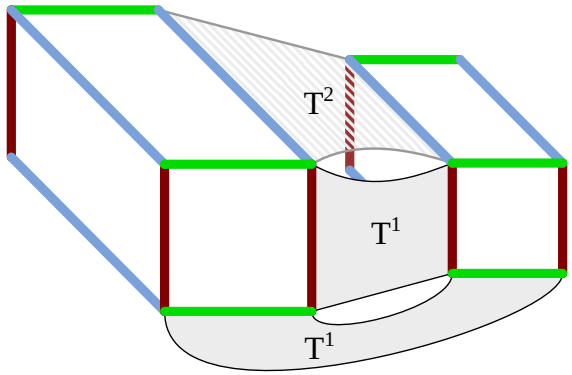
b) **OTT$_{11}$** (GW)



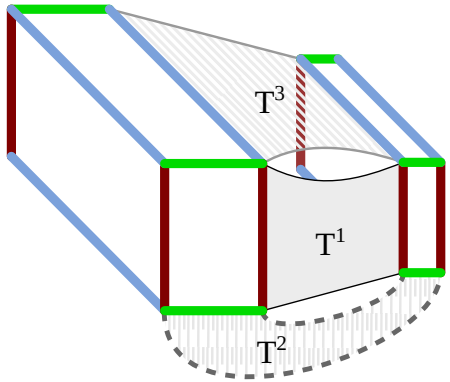
c) **OTT$_{12}$** (Co-OT)



d) **OTT$_{111}$** (triplets)

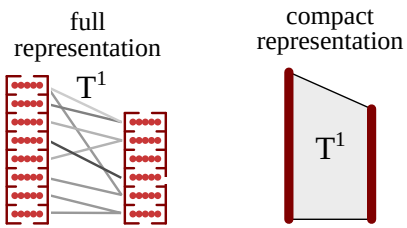


e) **OTT$_{112}$** (GW collections)

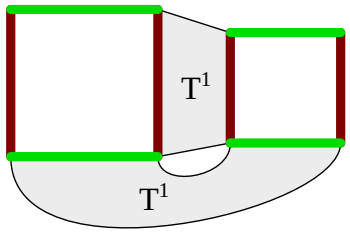


f) **OTT$_{123}$** (triCo-OT)

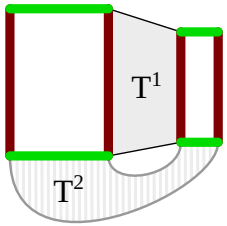
Motivation of Optimal Tensor Transport



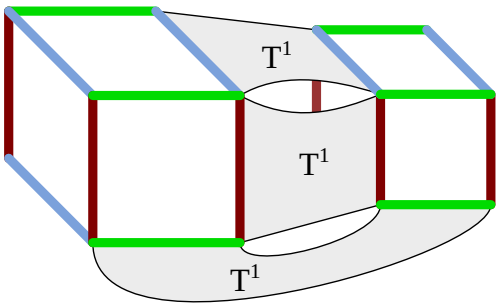
a) **OTT$_1$** (OT) *(F=5 features)*



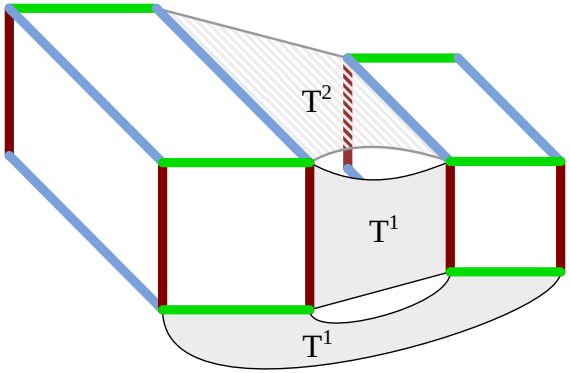
b) **OTT$_{11}$** (GW)



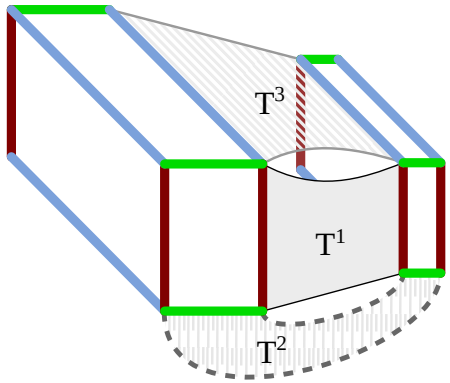
c) **OTT$_{12}$** (Co-OT)



d) **OTT$_{111}$** (triplets)



e) **OTT$_{112}$** (GW collections)

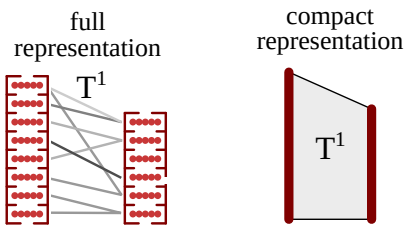


f) **OTT$_{123}$** (triCo-OT)

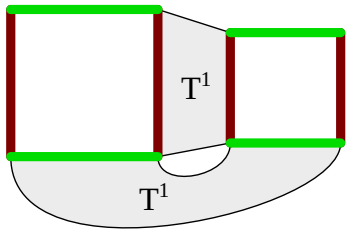
Optimal Transport

$$\mathrm{OT} = \min_{T^1 \in \Pi_{a^1 b^1}} \sum_{i_1=1}^{I_1} \sum_{k_1=1}^{K_1} \mathcal{L}(X_{i_1}, Y_{k_1})\, T^1_{i_1 k_1} \tag{8}$$

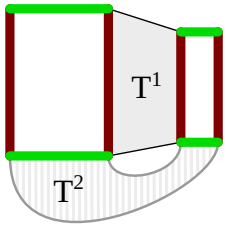
Motivation of Optimal Tensor Transport



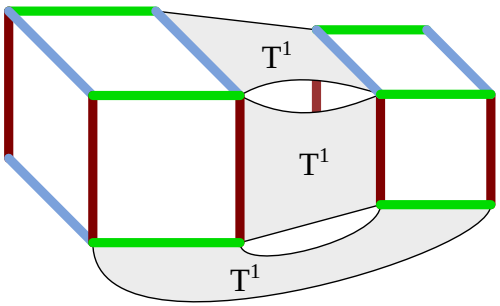
a) **OTT$_1$** (OT) *(F=5 features)*



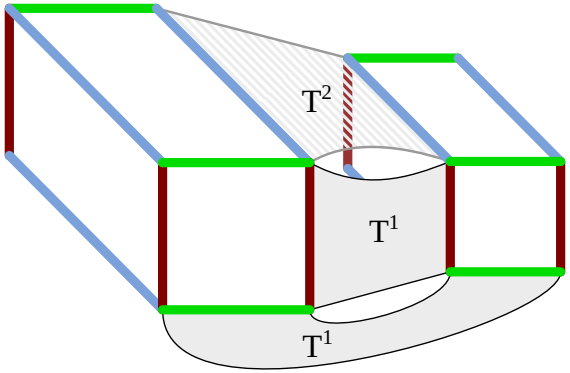
b) **OTT$_{11}$** (GW)



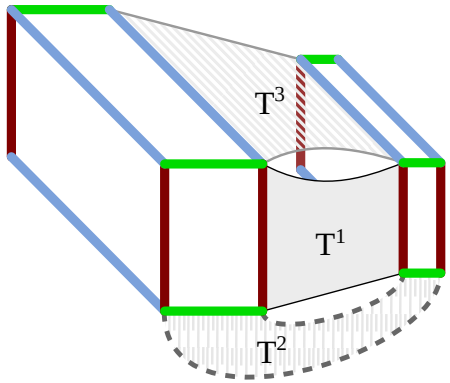
c) **OTT$_{12}$** (Co-OT)



d) **OTT$_{111}$** (triplets)



e) **OTT$_{112}$** (GW collections)

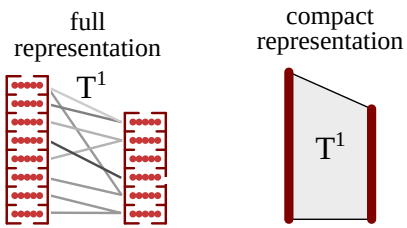


f) **OTT$_{123}$** (triCo-OT)

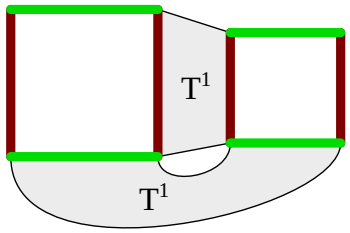
Gromov Wasserstein

$$\mathrm{GW} = \min_{T^1 \in \Pi_{a^1 b^1}} \sum_{i_1, i_2 = 1}^{I_1, I_2} \sum_{k_1, k_2 = 1}^{K_1, K_2} \mathcal{L}(X_{i_1 i_2}, Y_{k_1 k_2})\, T^1_{i_1 k_1}\, T^1_{i_2 k_2} \tag{8}$$

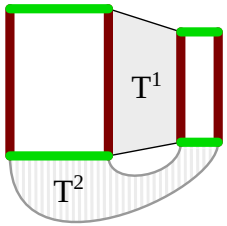
Motivation of Optimal Tensor Transport



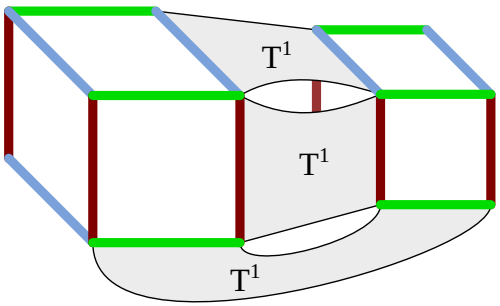
a) **OTT$_1$** (OT) *(F=5 features)*



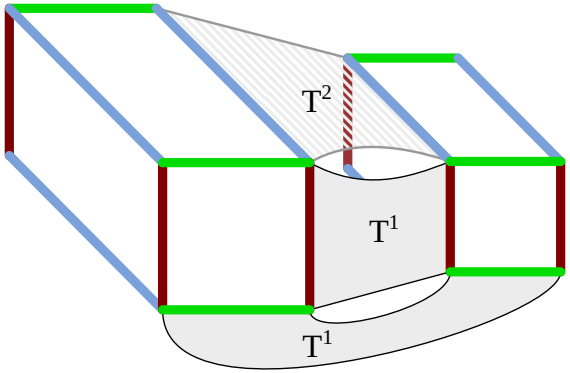
b) **OTT$_{11}$** (GW)



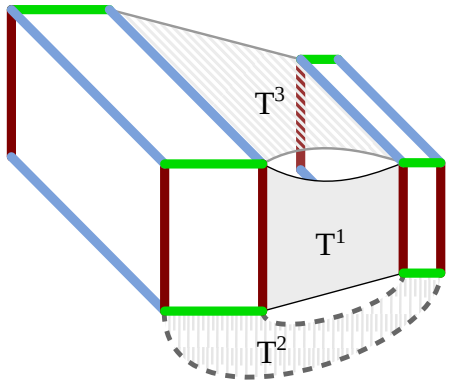
c) **OTT$_{12}$** (Co-OT)



d) **OTT$_{111}$** (triplets)



e) **OTT$_{112}$** (GW collections)

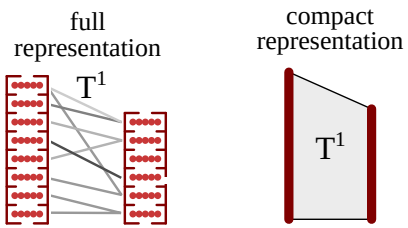


f) **OTT$_{123}$** (triCo-OT)

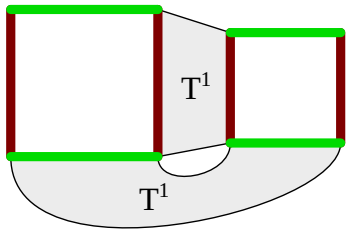
Co-Optimal Transport [Redko et al., 2020]

$$\text{Co-OT} = \min_{T^1 \in \Pi_{a^1 b^1}\, T^2 \in \Pi_{a^2 b^2}} \sum_{i_1, i_2 = 1}^{I_1, I_2} \sum_{k_1, k_2 = 1}^{K_1, K_2} \mathcal{L}(X_{i_1 i_2}, Y_{k_1 k_2})\, T^1_{i_1 k_1}\, T^2_{i_2 k_2} \qquad (8)$$

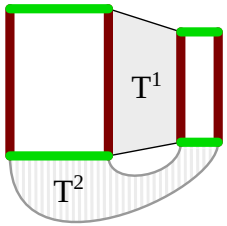
Motivation of Optimal Tensor Transport



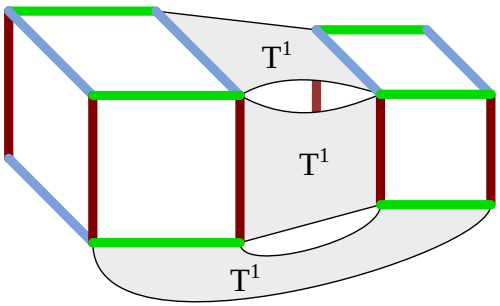
a) **OTT$_1$** (OT) *(F=5 features)*



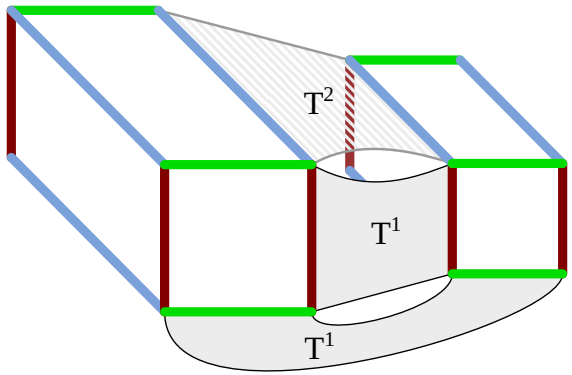
b) **OTT$_{11}$** (GW)



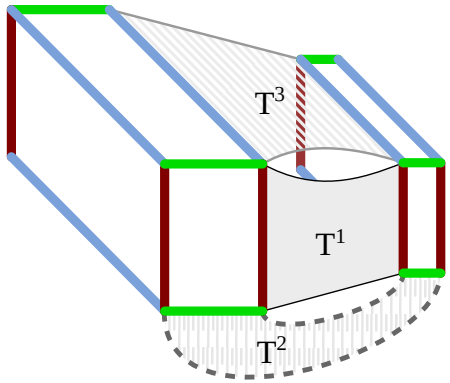
c) **OTT$_{12}$** (Co-OT)



d) **OTT$_{111}$** (triplets)



e) **OTT$_{112}$** (GW collections)

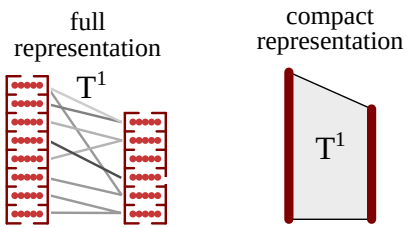


f) **OTT$_{123}$** (triCo-OT)

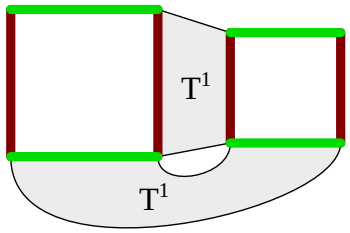
Optimal Tensor Transport with $(1, 1, 1)$

$$\min_{T^1 \in \Pi_{a^1 b^1}} \sum_{i_1,i_2,i_3=1}^{I_1,I_1,I_1} \sum_{k_1,k_2,k_3=1}^{K_1,K_1,K_1} \mathcal{L}(X_{i_1 i_2 i_3}, Y_{k_1 k_2 k_3})\, T^1_{i_1 k_1}\, T^1_{i_2 k_2}\, T^1_{i_3 k_3}$$

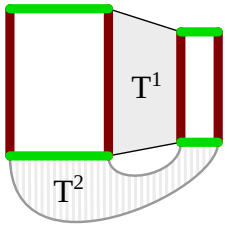
Motivation of Optimal Tensor Transport



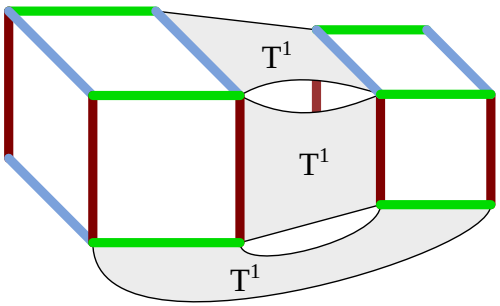
a) **OTT$_1$** (OT) *(F=5 features)*



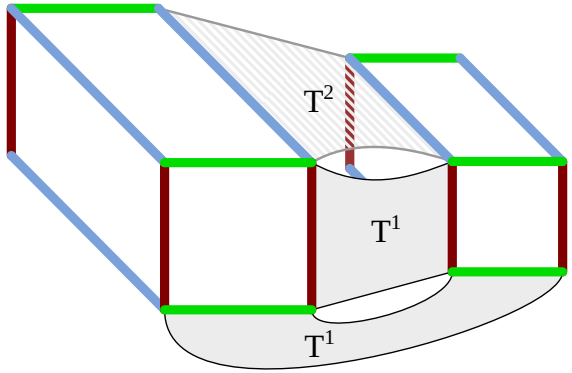
b) **OTT$_{11}$** (GW)



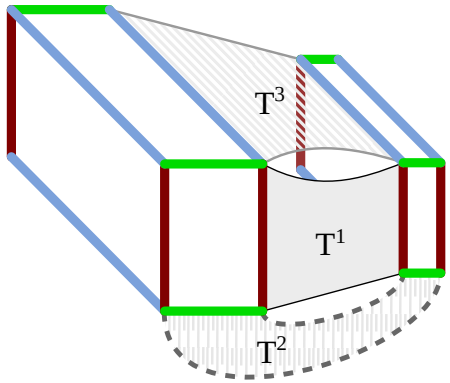
c) **OTT$_{12}$** (Co-OT)



d) **OTT$_{111}$** (triplets)



e) **OTT$_{112}$** (GW collections)

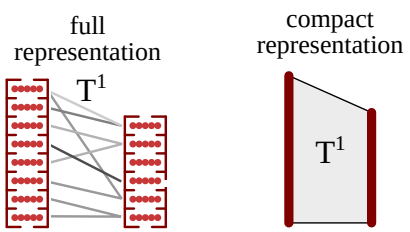


f) **OTT$_{123}$** (triCo-OT)

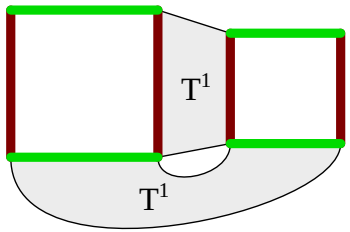
Optimal Tensor Transport with $(1, 1, 2)$

$$\min_{T^1 \in \Pi_{a^1 b^1}\, T^2 \in \Pi_{a^2 b^2}} \sum_{i_1, i_2, i_3 = 1}^{I_1, I_1, I_2} \sum_{k_1, k_2, k_3 = 1}^{K_1, K_1, K_2} \mathcal{L}(X_{i_1 i_2 i_3}, Y_{k_1 k_2 k_3})\, T^1_{i_1 k_1}\, T^1_{i_2 k_2}\, T^2_{i_3 k_3}$$

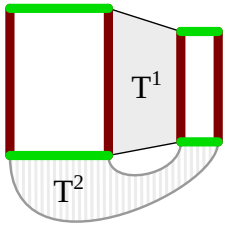
Motivation of Optimal Tensor Transport



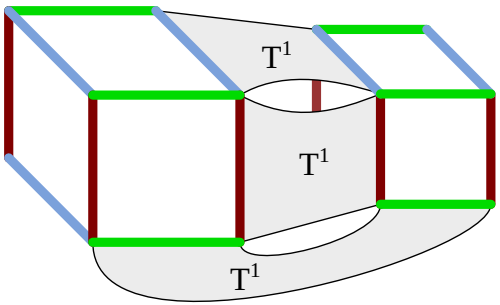
a) **OTT$_1$** (OT) *(F=5 features)*



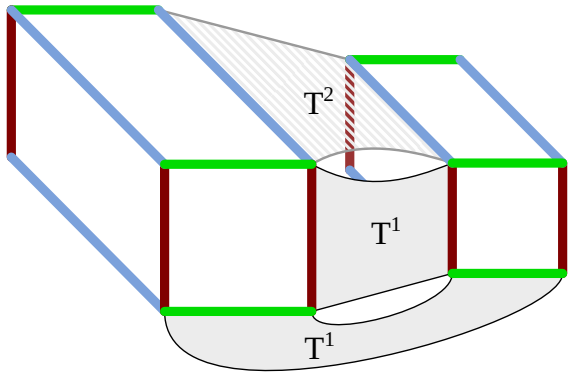
b) **OTT$_{11}$** (GW)



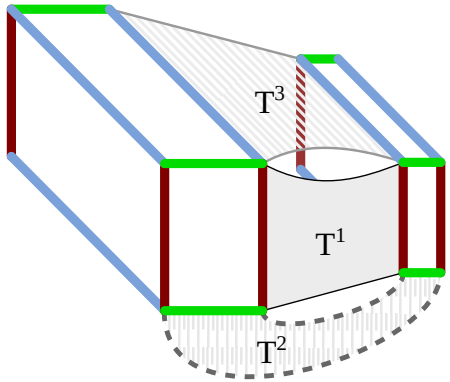
c) **OTT$_{12}$** (Co-OT)



d) **OTT$_{111}$** (triplets)



e) **OTT$_{112}$** (GW collections)



f) **OTT$_{123}$** (triCo-OT)

Optimal Tensor Transport with $(1, 2, 3)$

$$\forall e \min_{T^e \in \Pi_{a^e b^e}} \sum_{i_1,i_2,i_3=1}^{I_1,I_2,I_3} \sum_{k_1,k_2,k_3=1}^{K_1,K_2,K_3} \mathcal{L}(X_{i_1 i_2 i_3}, Y_{k_1 k_2 k_3})\, T^1_{i_1 k_1}\, T^2_{i_2 k_2}\, T^3_{i_3 k_3}$$

Mathematical formulations

Optimal Tensor Transport

$$\mathrm{OTT}_f(X, Y, (a^e)_{e\in[\![1,E]\!]}, (b^e)_{e\in[\![1,E]\!]}) = \min_{\forall e\ T^e\in\Pi_{a^e b^e}} \mathcal{E}_f\left(X, Y, (T^e)_{e\in[\![1,E]\!]}\right)$$

$$\mathcal{E}_f\left(X, Y, (T^e)_{e\in[\![1,E]\!]}\right) = \sum_{i_1,\dots,i_D=1}^{I_{f(1)},\dots,I_{f(D)}} \sum_{k_1,\dots,k_D=1}^{K_{f(1)},\dots,K_{f(D)}} \mathcal{L}(X_{i_1\dots i_D}, Y_{k_1\dots k_D}) \prod_{d=1}^{D} T^{f(d)}_{i_d k_d}$$

Stochastic Mirror Descent to solve the OTT problem

The gradient require N^{2D} operation

$$\nabla_{T^e}\mathcal{E}_f = \sum_{\{d'|f(d')=e\}} \mathbb{E}\left(\Lambda^{d'}\right)$$

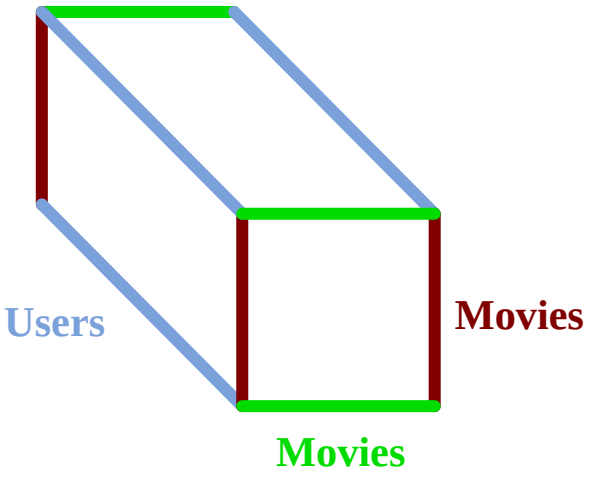
Algorithm 8 Alternated Stochastic Mirror Descent algorithm

Require: $(a^e)_{e\in[\![1,E]\!]}$, $(b^e)_{e\in[\![1,E]\!]}$, X, Y, $\mathcal{L}$, M, ϵ

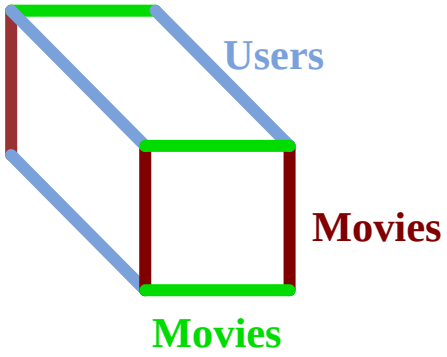
1: $\forall e \in [\![1, E]\!]$, $T^e = a^e b^{e\top}$
2: **for** s = 0 **to** S-1 **do**
3: **for** e = 1 **to** E **do**
4: $\widehat{\nabla_{T^e}\mathcal{E}_f} = M$ samples of the gradient
5: $T^e = \underset{T'\in\Pi(a^e,b^e)}{min} \left\langle \widehat{\nabla_{T^e}\mathcal{E}_f}, T' \right\rangle + \epsilon KL(T'||T^e)$
6: **end for**
7: **end for**

Interest of such formulation: Domain Adaptation

Domain Adaptation: predict the type of movies



Old movies



New movies

Datasets	Rdm	SVM	S-GWL	GW	Co-OT	OTT
AVG	50.0±3.3	58.6	63.8	64.9	68.2	**77.3**±3.1
AVG (best)	50.0±3.3	58.6	66.3	71.0	70.7	**78.9**±2.9

Conclusion

1. MLOT
   - ✓ Interest of **learning a ground metric** for a DA task.

Conclusion

1. MLOT
   - ✓ Interest of **learning a ground metric** for a DA task.
2. Minimax
   - ✓ General **cutting set** method for minimax OT problems
   - ✓ **Wasserstein stability** definition

Conclusion

1. MLOT
   - ✓ Interest of **learning a ground metric** for a DA task.
2. Minimax
   - ✓ General **cutting set** method for minimax OT problems
   - ✓ **Wasserstein stability** definition
3. Sampled GW
   - ✓ Fast approximation of the GW distance with **any losses**
   - ✓ Theoretical **convergence bound** to stationary points
   - ✓ **Very fast approximation** of the GW distance using the 1D OT fast solver

Conclusion

1. MLOT
   - ✓ Interest of **learning a ground metric** for a DA task.
2. Minimax
   - ✓ General **cutting set** method for minimax OT problems
   - ✓ **Wasserstein stability** definition
3. Sampled GW
   - ✓ Fast approximation of the GW distance with **any losses**
   - ✓ Theoretical **convergence bound** to stationary points
   - ✓ **Very fast approximation** of the GW distance using the 1D OT fast solver
4. OTT
   - ✓ New OT formulation to handle **tensors** of arbitrary orders
   - ✓ Adaptation of the previous solver to solve it in a reasonable time

Further works

Related to the contributions

1. Deep version of MLOT
2. Extend the use of the Wasserstein stability with a min minimax formulation
3. Convergence bound of SaGroW^{KL} to stationary points
4. Fused OTT

Further works

Related to the contributions

1. Deep version of MLOT
2. Extend the use of the Wasserstein stability with a min minimax formulation
3. Convergence bound of SaGroW^{KL} to stationary points
4. Fused OTT

Some perspectives for the OT's application to ML

1. GW $\Leftrightarrow$ Linear GAN
2. Under which condition on $\mathcal{L}$, GW has a sparse solution ?

Biblio I

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1988). Network flows.

Brandes, U., Gaertler, M., and Wagner, D. (2003). Experiments on graph clustering algorithms. In *European Symposium on Algorithms*.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2017). Optimal transport for domain adaptation. *PAMI*.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*.

Dhouib, S., Redko, I., Kerdoncuff, T., Emonet, R., and Sebban, M. (2020). A swiss army knife for minimax optimal transport. In *International Conference on Machine Learning*.

Frank, M., Wolfe, P., et al. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*.

Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*.

Kantorovich, L. (1942). On the translocation of masses. *Doklady of the Academy of Sciences of the USSR*, 37:199–201.

Biblio II

Kerdoncuff, T., Emonet, R., and Sebban, M. (2020). Metric learning in optimal transport for domain adaptation. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2162–2168. International Joint Conferences on Artificial Intelligence Organization. Main track.

Kerdoncuff, T., Emonet, R., and Sebban, M. (2021). Sampled gromov wasserstein. *Machine Learning*.

Memoli, F. (2007). On the use of Gromov-Hausdorff Distances for Shape Comparison. In Botsch, M., Pajarola, R., Chen, B., and Zwicker, M., editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie royale des sciences de Paris*.

Mutapcic, A. and Boyd, S. P. (2009). Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods and Software*.

Peyré, G., Cuturi, M., and Solomon, J. (2016). Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*.

Biblio III

Reddi, S. J., Sra, S., Póczos, B., and Smola, A. (2016). Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*.

Redko, I., Vayer, T., Flamary, R., and Courty, N. (2020). Co-optimal transport. In *NeurIPS 2020-Thirty-four Conference on Neural Information Processing Systems*.

Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*.

Villani, C. (2008). *Optimal transport: old and new*.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *JMLR*.

Xu, H., Luo, D., and Carin, L. (2019). Scalable gromov-wasserstein learning for graph partitioning and matching. In *Advances in neural information processing systems*.

Zhong, E., Fan, W., Yang, Q., Verscheure, O., and Ren, J. (2010). Cross validation framework to choose amongst models and datasets for transfer learning. In *ECML PKDD*.

Conclusions

- Problem:
  - We tackle the challenging unsupervised DA problem
  - OT for DA always relies on the euclidean distance
- ✓ Solutions:
  - We provide a metric dependent bound on the target error
  - We use labeled sources to learn a class-informed metric
  - We project, independently, the source and target point into subspaces
  - We show a relation between OT and PCA, and use PCA as initialization
- ✓ Results:
  - Competitive against various baselines with a Reverse Validation method
  - Improve the performance of OTDA for different set of hyperparameters

Continuous GW