Stochastic Optimal Transport with applications to flow cytometry data

Paul Freulon

Rencontre Poitiers-Bordeaux, December 2020

Université de Bordeaux

Context

Flow cytometry data

Quantifying cellular markers in a biological sample (e.g. blood draw) cell-by-cell.

Context

Flow cytometry data

Quantifying cellular markers in a biological sample (e.g. blood draw) cell-by-cell.

- The biological markers are stained.
- The light emitted by a maker indicates whether the marker is present or not on the cell.

Modelling

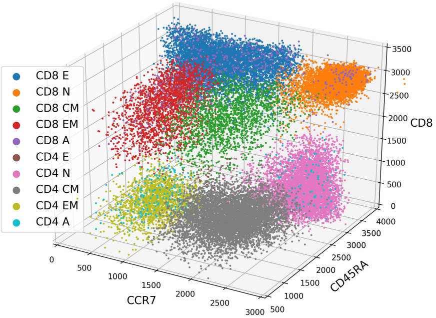
- The observation $X_i \in \mathbb{R}^d$ corresponds to the measures on the i^{th} cell.
- For $m \in \{1, ..., d\}$, the coefficient $X_i^{(m)}$ corresponds to the light emitted by the biological marker m.
- In a data set $X_1, ..., X_I$, the number of observation range from 10 000 to 200 000.

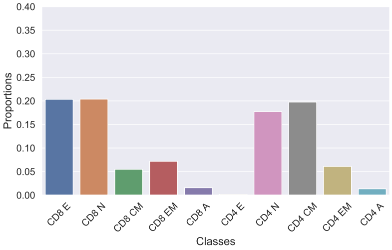
Modelling

- The observation $X_i \in \mathbb{R}^d$ corresponds to the measures on the i^{th} cell.
- For $m \in \{1, ..., d\}$, the coefficient $X_i^{(m)}$ corresponds to the light emitted by the biological marker m.
- In a data set $X_1, ..., X_I$, the number of observation range from 10 000 to 200 000.

CCR7	CD4	CD45RA	CD3	HLADR	CD38	CD8
717.3	1146.5	3094.8	2526.3	1333.1	1510.2	3203.7

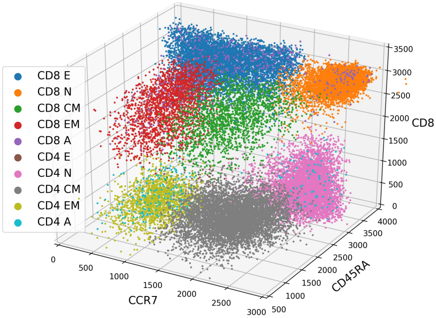
Table 1: Cytometry measurement for one cell. $d = 7$.

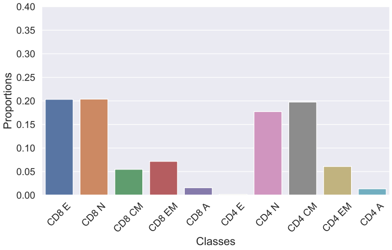




Objective

Quantify relative abundance of cell types within the sample.





Objective

Quantify relative abundance of cell types within the sample.

Application

clinical practice: Monitor human disease and response to therapy.

Data analysis

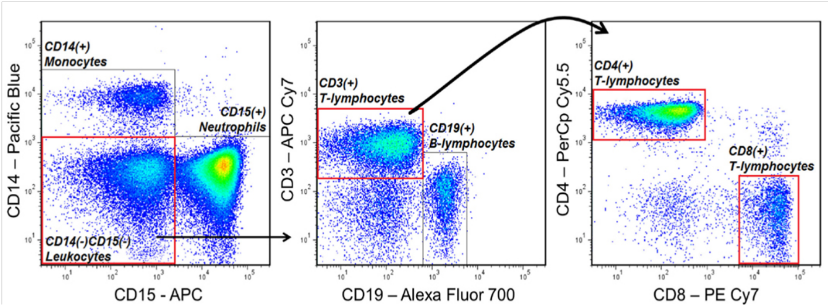


Figure 2: Front. Immunol., 27 July 2015.

Manual gating

Drawbacks: Time consuming, expensive and poorly reproducible.

Automated methods

Unsupervised methods

- Kmeans
- Hierarchical clustering
- Mixture models

Automated methods

Unsupervised methods

- Kmeans
- Hierarchical clustering
- Mixture models

Supervised methods

- Deep learning
- Quadratic discriminant analysis
- Random Forest

Automated methods

Unsupervised methods

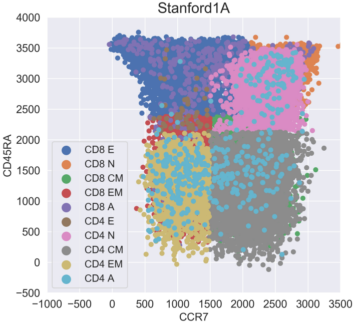
- Kmeans
- Hierarchical clustering
- Mixture models

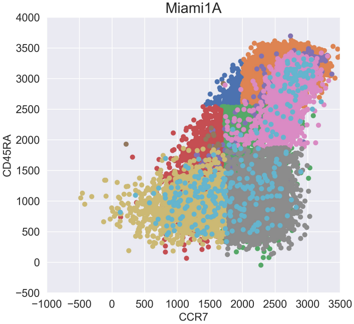
Supervised methods

- Deep learning
- Quadratic discriminant analysis
- Random Forest

Manual gating is still the benchmark methods.

Challenges of the automated analysis

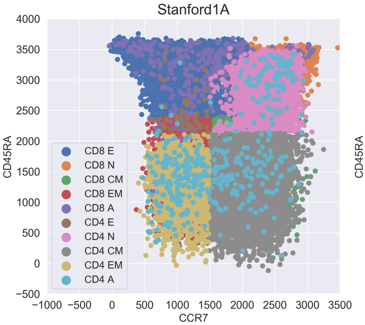


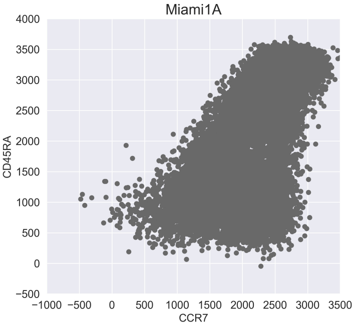


Technical variability

Two samples analysed from the same patient. Cytometry measurements were performed in two different laboratories.

Challenges of the automated analysis





- One gated data set.
- Derive the class proportions in an unclassified data set.

Wasserstein Distance

Let α and β two probability measures on $\mathbb{R}^d$ with finite second moment.

Let $\Pi(\alpha, \beta)$ be the set of probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals α and β.

Wasserstein Distance

Let α and β two probability measures on $\mathbb{R}^d$ with finite second moment.

Let $\Pi(\alpha, \beta)$ be the set of probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals α and β.

Definition

The Wasserstein distance between α and β is defined as

$$W_2^2(\alpha, \beta) = \min_{\pi \in \Pi(\alpha, \beta)} \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x, y) d\pi(x, y). \tag{1}$$

where $c(x, y) = ||x - y||_2^2$.

Discrete Setting

If $\alpha = \sum_i^I a_i \delta_{x_i}$ and $\beta = \sum_{j=1}^J b_j \delta_{y_j}$ are two discrete probability distributions on $\mathbb{R}^d$ the Wasserstein distance reads:

$$W_2^2(\alpha, \beta) = \min_{P \in U(a,b)} \sum_{i,j} C_{i,j} P_{i,j} \tag{2}$$

where $C_{i,j} = ||x_i - y_j||_2^2$.

Wasserstein Distance

Discrete Setting

If $\alpha = \sum_i^I a_i \delta_{x_i}$ and $\beta = \sum_{j=1}^J b_j \delta_{y_j}$ are two discrete probability distributions on $\mathbb{R}^d$ the Wasserstein distance reads:

$$W_2^2(\alpha, \beta) = \min_{P \in U(a,b)} \sum_{i,j} C_{i,j} P_{i,j} \tag{2}$$

where $C_{i,j} = ||x_i - y_j||_2^2$.

Computational Cost

Suppose α and β are two measures with equal size N.

- Requires to store a $N \times N$ matrix.
- Linear programming problem.
- $O(N^3 \log(N))$ operations required.

Entropic regularization (M.Cuturi 2013)

Regularized Wasserstein Distance

For α and β two probability measures the regularized Wasserstein distance is defined as:

$$W^{\varepsilon}(\alpha,\beta) = \min_{\pi \in \Pi(\alpha,\beta)} \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x,y) d\pi(x,y) + \varepsilon H(\pi). \tag{3}$$

where $\varepsilon > 0$ and
$H(\pi) = \int_{\mathbb{R}^d \times \mathbb{R}^d} \log\left(\left(\frac{d\pi}{d\alpha \otimes \beta}(x,y)\right) - 1\right) d\pi(x,y)$

Entropic regularization (M.Cuturi 2013)

Regularized Wasserstein Distance

For α and β two probability measures the regularized Wasserstein distance is defined as:

$$W^{\varepsilon}(\alpha,\beta) = \min_{\pi\in\Pi(\alpha,\beta)} \int_{\mathbb{R}^d\times\mathbb{R}^d} c(x,y)d\pi(x,y) + \varepsilon H(\pi). \qquad (3)$$

where $\varepsilon > 0$ and
$H(\pi) = \int_{\mathbb{R}^d\times\mathbb{R}^d} \log\left(\left(\frac{d\pi}{d\alpha\otimes\beta}(x,y)\right) - 1\right) d\pi(x,y)$

Dual problem

$$\begin{aligned} W^{\varepsilon}(\alpha,\beta) = \sup_{u,v\in\mathcal{C}(\mathbb{R}^d)} &\int u(x)d\alpha(x) + \int v(y)d\beta(y) \\ &- \varepsilon \int e^{\frac{u(x)+v(y)-c(x,y)}{\varepsilon}} d\alpha\otimes\beta(x,y) \end{aligned} \qquad (4)$$

Dual problem in a discrete setting

$$W^{\varepsilon}(\alpha, \beta) = \max_{u \in \mathbb{R}^I, v \in \mathbb{R}^J} \langle u, a \rangle + \langle v, b \rangle - \varepsilon \langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b \rangle. \quad (5)$$

Dual problem in a discrete setting

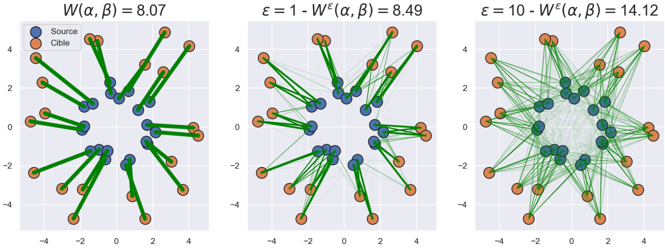
$$W^{\varepsilon}(\alpha, \beta) = \max_{u \in \mathbb{R}^I, v \in \mathbb{R}^J} \langle u, a \rangle + \langle v, b \rangle - \varepsilon \langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b \rangle. \qquad (5)$$

Block coordinate ascent strategy $\rightarrow$ Sinkhorn algorithm.

Dual problem in a discrete setting

$$W^{\varepsilon}(\alpha, \beta) = \max_{u \in \mathbb{R}^{I}, v \in \mathbb{R}^{J}} \langle u, a \rangle + \langle v, b \rangle - \varepsilon \langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b \rangle. \qquad (5)$$

Block coordinate ascent strategy $\rightarrow$ Sinkhorn algorithm. In the case where $I = J = N$, computation of a solution in $O(N^2 \log(N))$ operations.


$W(\alpha, \beta) = 8.07$
$\varepsilon = 1$ - $W^{\varepsilon}(\alpha, \beta) = 8.49$
$\varepsilon = 10$ - $W^{\varepsilon}(\alpha, \beta) = 14.12$
Source
Cible

Maximizing over a single potential $u \in \mathbb{R}^I$

For $u \in \mathbb{R}^I$, the unique maximizer of the function

$$F(v) = \langle u, a \rangle + \langle v, b \rangle - \varepsilon \langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b \rangle$$

is the smoothed c-transform of u:

$$u_{c,\varepsilon}(y_j) = -\varepsilon \log \left(\sum_{i=1}^{I} e^{\frac{u(x_i) - c(x_i, y_j)}{\varepsilon}} a_i \right)$$

From Dual to Semi-Dual

Maximizing over a single potential $u \in \mathbb{R}^I$

For $u \in \mathbb{R}^I$, the unique maximizer of the function

$$F(v) = \langle u, a \rangle + \langle v, b \rangle - \varepsilon \langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b \rangle$$

is the smoothed c-transform of u:

$$u_{c,\varepsilon}(y_j) = -\varepsilon \log \left(\sum_{i=1}^{I} e^{\frac{u(x_i) - c(x_i, y_j)}{\varepsilon}} a_i \right)$$

by plug-in $u_{c,\varepsilon}$ in the dual formulation, we got :

$W^\varepsilon(\alpha, \beta) =$

From Dual to Semi-Dual

Maximizing over a single potential $u \in \mathbb{R}^I$

For $u \in \mathbb{R}^I$, the unique maximizer of the function

$$F(v) = \langle u, a \rangle + \langle v, b \rangle - \varepsilon \langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b \rangle$$

is the smoothed c-transform of u:

$$u_{c,\varepsilon}(y_j) = -\varepsilon \log \left(\sum_{i=1}^{I} e^{\frac{u(x_i) - c(x_i, y_j)}{\varepsilon}} a_i \right)$$

by plug-in $u_{c,\varepsilon}$ in the dual formulation, we got :

$$W^\varepsilon(\alpha, \beta) = \max_{u \in \mathbb{R}^I} \langle u, a \rangle + \langle u_{c,\varepsilon}, b \rangle - \varepsilon \langle e^{\frac{u \oplus u_{c,\varepsilon} - C}{\varepsilon}}, a \otimes b \rangle$$

From Dual to Semi-Dual

Maximizing over a single potential $u \in \mathbb{R}^I$

For $u \in \mathbb{R}^I$, the unique maximizer of the function

$$F(v) = \langle u, a \rangle + \langle v, b \rangle - \varepsilon \langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b \rangle$$

is the smoothed c-transform of u:

$$u_{c,\varepsilon}(y_j) = -\varepsilon \log \left(\sum_{i=1}^{I} e^{\frac{u(x_i) - c(x_i, y_j)}{\varepsilon}} a_i \right)$$

by plug-in $u_{c,\varepsilon}$ in the dual formulation, we got :

$$\begin{aligned} W^\varepsilon(\alpha, \beta) &= \max_{u \in \mathbb{R}^I} \langle u, a \rangle + \langle u_{c,\varepsilon}, b \rangle - \varepsilon \langle e^{\frac{u \oplus u_{c,\varepsilon} - C}{\varepsilon}}, a \otimes b \rangle \\ &= \max_{u \in \mathbb{R}^I} \langle u, a \rangle + \langle u_{c,\varepsilon}, b \rangle - \varepsilon \sum_{j=1}^{J} e^{\frac{u_{c,\varepsilon}(y_j)}{\varepsilon}} \langle e^{\frac{u - C(., y_j)}{\varepsilon}}, a \rangle b_j. \end{aligned}$$

From Dual to Semi-Dual

Maximizing over a single potential $u \in \mathbb{R}^I$

For $u \in \mathbb{R}^I$, the unique maximizer of the function

$$F(v) = \langle u, a\rangle + \langle v, b\rangle - \varepsilon\langle e^{\frac{u \oplus v - C}{\varepsilon}}, a \otimes b\rangle$$

is the smoothed c-transform of u:

$u_{c,\varepsilon}(y_j) = -\varepsilon \log\left(\sum_{i=1}^{I} e^{\frac{u(x_i) - c(x_i, y_j)}{\varepsilon}} a_i\right)$

by plug-in $u_{c,\varepsilon}$ in the dual formulation, we got :

$$\begin{aligned} W^\varepsilon(\alpha, \beta) &= \max_{u \in \mathbb{R}^I} \langle u, a\rangle + \langle u_{c,\varepsilon}, b\rangle - \varepsilon\langle e^{\frac{u \oplus u_{c,\varepsilon} - C}{\varepsilon}}, a \otimes b\rangle \\ &= \max_{u \in \mathbb{R}^I} \langle u, a\rangle + \langle u_{c,\varepsilon}, b\rangle - \varepsilon \sum_{j=1}^{J} e^{\frac{u_{c,\varepsilon}(y_j)}{\varepsilon}} \langle e^{\frac{u - C(.,y_j)}{\varepsilon}}, a\rangle b_j. \end{aligned}$$

And $e^{\frac{u_{c,\varepsilon}(y_j)}{\varepsilon}} = \left(\langle e^{\frac{u - C(.,y_j)}{\varepsilon}}, a\rangle\right)^{-1}$

Thus

Semi Dual Formulation

$$W^{\varepsilon}(\alpha, \beta) = \max_{u \in \mathbb{R}^{I}} \langle u, a \rangle + \langle u_{c,\varepsilon}, b \rangle - \varepsilon$$

In the case where $\alpha = \sum_{i=1}^{I} a_i \delta_{x_i}$ and $\beta = \sum_{j=1}^{J} b_j \delta_{y_j}$ are discrete.

Thus

Semi Dual Formulation

$$W^\varepsilon(\alpha,\beta) = \max_{u\in\mathbb{R}^I} \langle u, a\rangle + \langle u_{c,\varepsilon}, b\rangle - \varepsilon$$

In the case where $\alpha = \sum_{i=1}^I a_i\delta_{x_i}$ and $\beta = \sum_{j=1}^J b_j\delta_{y_j}$ are discrete. $W^\varepsilon(\alpha,\beta)$ also reads :

$$W^\varepsilon(\alpha,\beta) = \max_{u\in\mathbb{R}^I} \sum_{i=1}^I u_i a_i + \int_{\mathbb{R}^d} u_{c,\varepsilon} d\beta - \varepsilon$$

And

$$u_{c,\varepsilon}(y) = -\varepsilon \log\left(\sum_{i=1}^I e^{\frac{u(x_i)-c(x_i,y)}{\varepsilon}} a_i\right)$$

Thus

Semi Dual Formulation

$$W^{\varepsilon}(\alpha, \beta) = \max_{u \in \mathbb{R}^I} \langle u, a \rangle + \langle u_{c,\varepsilon}, b \rangle - \varepsilon$$

In the case where $\alpha = \sum_{i=1}^{I} a_i \delta_{x_i}$ and $\beta = \sum_{j=1}^{J} b_j \delta_{y_j}$ are discrete. $W^{\varepsilon}(\alpha, \beta)$ also reads :

$$W^{\varepsilon}(\alpha, \beta) = \max_{u \in \mathbb{R}^I} \sum_{i=1}^{I} u_i a_i + \int_{\mathbb{R}^d} u_{c,\varepsilon} d\beta - \varepsilon$$

And

$$u_{c,\varepsilon}(y) = -\varepsilon \log \left(\sum_{i=1}^{I} e^{\frac{u(x_i) - c(x_i, y)}{\varepsilon}} a_i \right)$$

$\rightarrow$ Expectation formulation of the regularized Wasserstein distance.

Stochastic optimal transport

Let β be any probability measure and $\alpha = \sum_{i=1}^{I} a_i \delta_{x_i}$.

Proposition (Genevay, Cuturi, Peyré and Bach (2016))

let $\varepsilon \geq 0$,

$$W_\varepsilon(\alpha, \beta) = \max_{u \in \mathbb{R}^I} \mathbb{E}[g_\varepsilon(Y, u)] \tag{6}$$

- Y is a random variable with distribution β.
- $g_\varepsilon(y, u) = \sum_{i=1}^{I} u_i a_i + u_{c,\varepsilon}(y) - \varepsilon$ is easy to compute for all $y \in \mathbb{R}^d$, and all $u \in \mathbb{R}^I$

Stochastic optimal transport

Let β be any probability measure and $\alpha = \sum_{i=1}^{I} a_i \delta_{x_i}$.

Proposition (Genevay, Cuturi, Peyré and Bach (2016))

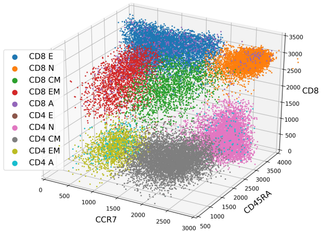
let $\varepsilon \geq 0$,

$$W_\varepsilon(\alpha, \beta) = \max_{u \in \mathbb{R}^I} \mathbb{E}[g_\varepsilon(Y, u)] \tag{6}$$

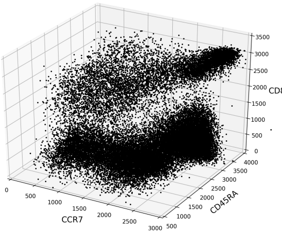
- Y is a random variable with distribution β.
- $g_\varepsilon(y, u) = \sum_{i=1}^{I} u_i a_i + u_{c,\varepsilon}(y) - \varepsilon$ is easy to compute for all $y \in \mathbb{R}^d$, and all $u \in \mathbb{R}^I$

- Stochastic optimization techniques can be applied.
- No need to store the full cost matrix.

Application to flow cytometry data



(a) Stanford Patient 1



(b) Stanford Patient 3

Domain adaptation (R.Flammary et al. (2019))

Framework

- The source distribution α is a mixture model.
- The target distribution β is a mixture model.

Domain adaptation (R.Flammary et al. (2019))

Framework

- The source distribution α is a mixture model.
- The target distribution β is a mixture model.

Idea

Re-weight the source distribution in order to reduce the Wasserstein distance $W(\alpha, \beta)$ between the source and target distribution.

Domain adaptation (R.Flammary et al. (2019))

Framework

- The source distribution α is a mixture model.
- The target distribution β is a mixture model.

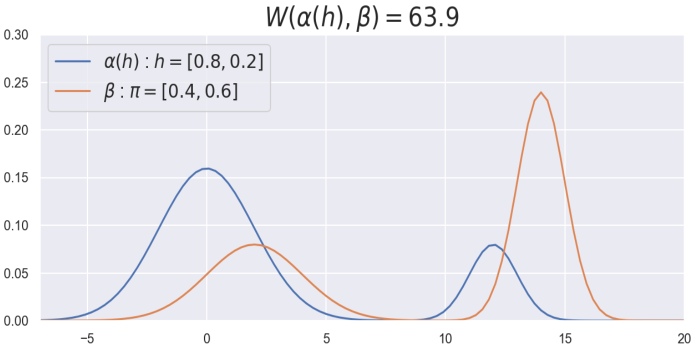
Idea

Re-weight the source distribution in order to reduce the Wasserstein distance $W(\alpha, \beta)$ between the source and target distribution.

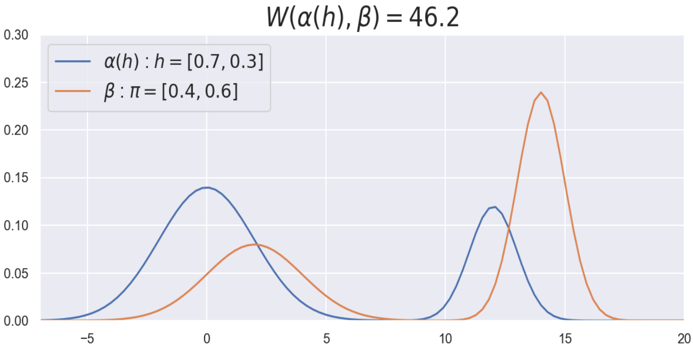
Goal

Estimation of the weights of the mixture π in the target distribution β.

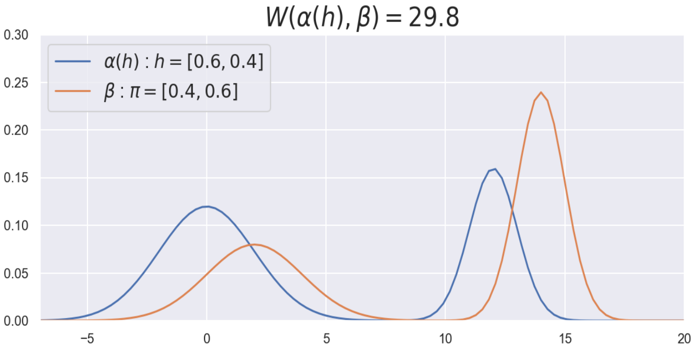
Domain adaptation (R.Flammary et al. (2019))



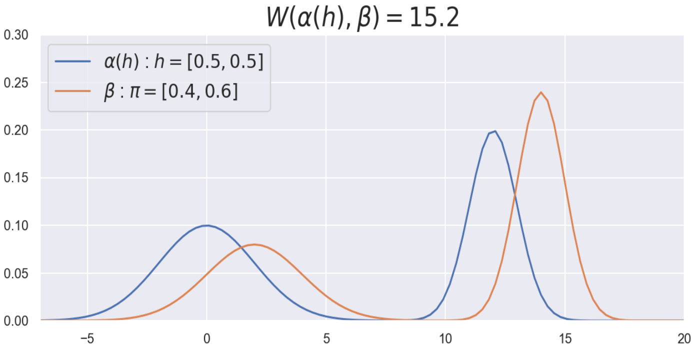
Domain adaptation (R.Flammary et al. (2019))



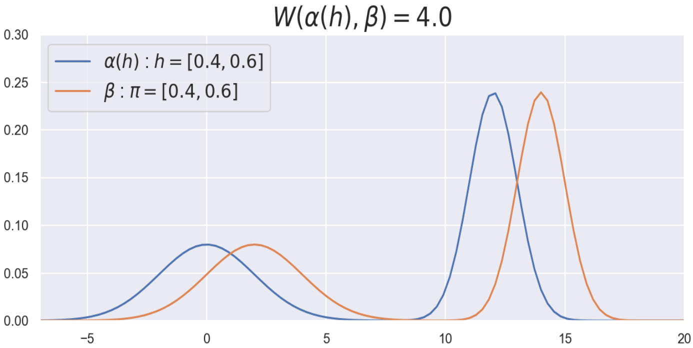
Domain adaptation (R.Flammary et al. (2019))



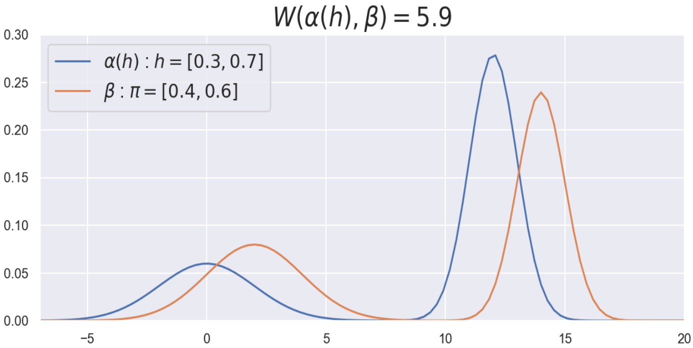
Domain adaptation (R.Flammary et al. (2019))



Domain adaptation (R.Flammary et al. (2019))



Domain adaptation (R.Flammary et al. (2019))



- Target measure : $\beta = \frac{1}{J}\sum_{j=1}^{J} \delta_{Y_j}$
- Source measure : $\alpha = \frac{1}{I}\sum_{i=1}^{I} \delta_{X_i}$

- Target measure : $\beta = \frac{1}{J}\sum_{j=1}^{J}\delta_{Y_j}$
- Source measure : $\alpha = \frac{1}{I}\sum_{i=1}^{I}\delta_{X_i}$
- Classification available for the source data set
  $\rightarrow \alpha = \sum_{k=1}^{K}\alpha_k$

- Target measure : $\beta = \frac{1}{J}\sum_{j=1}^{J}\delta_{Y_j}$
- Source measure : $\alpha = \frac{1}{I}\sum_{i=1}^{I}\delta_{X_i}$
- Classification available for the source data set
  $\rightarrow \alpha = \sum_{k=1}^{K}\alpha_k$

Re-weighting of the source data

For $h = (h_1, ..., h_K) \in \Sigma_K$, the measure α re weighted by h is:

$$\alpha(h) = \sum_{k=1}^{K} h_k \alpha_k \tag{7}$$

- Target measure : $\beta = \frac{1}{J}\sum_{j=1}^{J}\delta_{Y_j}$
- Source measure : $\alpha = \frac{1}{I}\sum_{i=1}^{I}\delta_{X_i}$
- Classification available for the source data set
  $\rightarrow \alpha = \sum_{k=1}^{K}\alpha_k$

Re-weighting of the source data

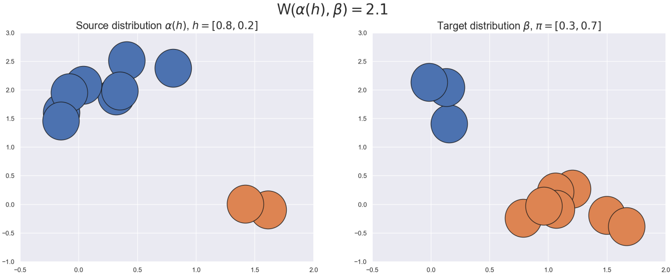
For $h = (h_1, ..., h_K) \in \Sigma_K$, the measure α re weighted by h is:

$$\alpha(h) = \sum_{k=1}^{K} h_k \alpha_k \tag{7}$$

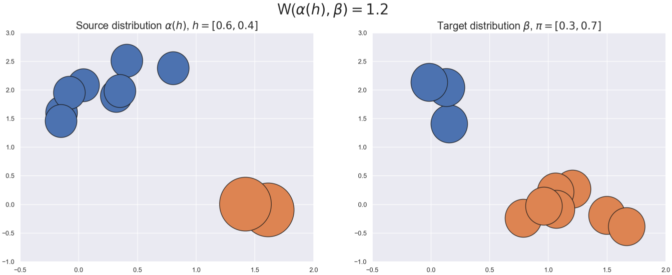
Estimation of the class proportions in the target data set:

$$\hat{\pi} \in \operatorname*{arg\,min}_{h \in \Sigma_K} W^{\varepsilon}(\alpha(h), \beta) \tag{8}$$

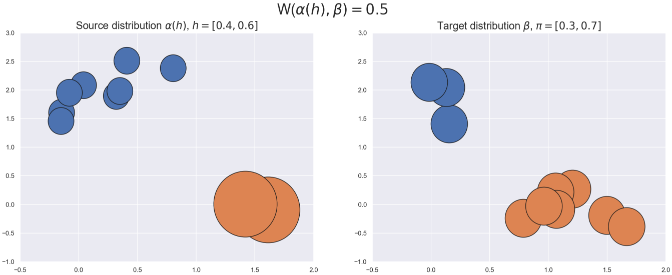
Illustration



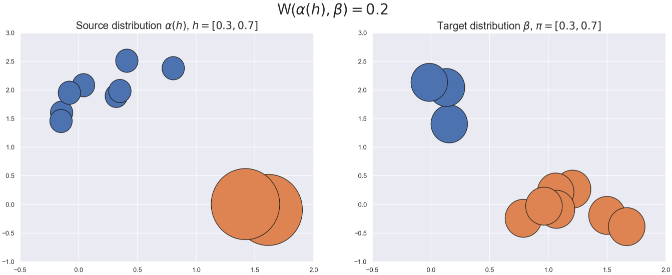
Illustration

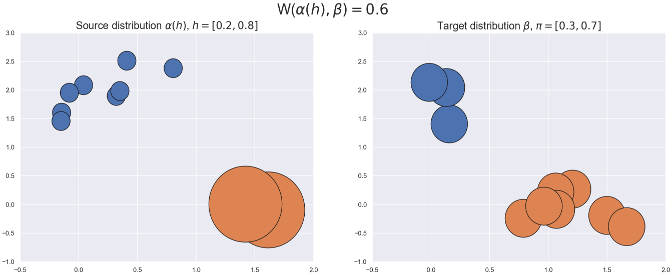


Illustration



Illustration





W(α(h), β) = 0.6
Source distribution α(h), h = [0.2, 0.8]
Target distribution β, π = [0.3, 0.7]

Descent Ascent procedure

$$\min_{h \in \Sigma_K} W^{\varepsilon}(\alpha(h), \beta)$$

Descent Ascent procedure

$$\min_{h \in \Sigma_K} W^{\varepsilon}(\alpha(h), \beta) = \min_{h \in \Sigma_K} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, h)], \tag{9}$$

where Y is a random variable with distribution β, and $g_{\varepsilon}(y, u, h)$ is easy to compute.

Descent Ascent procedure

$$\min_{h\in\Sigma_K} W^{\varepsilon}(\alpha(h),\beta) = \min_{h\in\Sigma_K}\max_{u\in\mathbb{R}^I}\mathbb{E}[g_\varepsilon(Y,u,h)], \tag{9}$$

where Y is a random variable with distribution β, and $g_\varepsilon(y,u,h)$ is easy to compute.

- re-parameterize problem (9) with a soft max function σ
- $\sigma(z)_l = \frac{\exp(z_l)}{\sum_{k=1}^{K}\exp(z_k)}$.
- Avoid projecting h on the simplex Σ_K

Descent Ascent procedure

$$\min_{h \in \Sigma_K} W^{\varepsilon}(\alpha(h), \beta) = \min_{h \in \Sigma_K} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, h)], \tag{9}$$

where Y is a random variable with distribution β, and $g_{\varepsilon}(y, u, h)$ is easy to compute.

- re-parameterize problem (9) with a soft max function σ
- $\sigma(z)_l = \frac{\exp(z_l)}{\sum_{k=1}^{K} \exp(z_k)}$.
- Avoid projecting h on the simplex Σ_K

New minimization problem:

$$\min_{z \in \mathbb{R}^K} W^{\varepsilon}(\alpha(\sigma(z)), \beta) = \min_{z \in \mathbb{R}^K} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))] \tag{10}$$

Objective function:

$$F : z \mapsto W^{\varepsilon}(\alpha(\sigma(z)), \beta) = \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$$

Objective function:

$$F : z \mapsto W^{\varepsilon}(\alpha(\sigma(z)), \beta) = \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$$

- Gradient computation: for $z \in \mathbb{R}^K$ as $\nabla_z F(z) = (\Gamma J_{\sigma}(z))^T u_z^*$.

Objective function:

$$F : z \mapsto W^{\varepsilon}(\alpha(\sigma(z)), \beta) = \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$$

- Gradient computation: for $z \in \mathbb{R}^K$ as $\nabla_z F(z) = (\Gamma J_{\sigma}(z))^T u_z^*$.
- u_z^* is a maximizer of the function $u \mapsto \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$

Objective function:

$$F : z \mapsto W^{\varepsilon}(\alpha(\sigma(z)), \beta) = \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$$

- Gradient computation: for $z \in \mathbb{R}^K$ as $\nabla_z F(z) = (\Gamma J_{\sigma}(z))^T u_z^*$.
- u_z^* is a maximizer of the function $u \mapsto \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$
- $\Gamma \in \mathbb{R}^{I \times K}$: linear operator.
- $J_{\sigma}(z)$: Jacobian matrix of the soft max function σ.

Objective function:

$$F : z \mapsto W^{\varepsilon}(\alpha(\sigma(z)), \beta) = \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$$

- Gradient computation: for $z \in \mathbb{R}^K$ as $\nabla_z F(z) = (\Gamma J_{\sigma}(z))^T u_z^*$.
- u_z^* is a maximizer of the function $u \mapsto \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$
- $\Gamma \in \mathbb{R}^{I \times K}$: linear operator.
- $J_{\sigma}(z)$: Jacobian matrix of the soft max function σ.

for $z \in \mathbb{R}^K$, we need to solve $\max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$.

$\rightarrow$ Stochastic optimal transport computation.

Approximation of $\nabla_z F(z)$

Robbins-Monro algorithm to approximate u_z^* with U_z. We get a stochastic approximation of $\nabla_z F(z)$ with:

$$\hat{\omega}(z) = (\Gamma J_\sigma(z))^T U_z$$

Approximation of $\nabla_z F(z)$

Robbins-Monro algorithm to approximate u_z^* with U_z. We get a stochastic approximation of $\nabla_z F(z)$ with:

$$\hat{\omega}(z) = (\Gamma J_\sigma(z))^T U_z$$

Descent-Ascent Algorithm

- Gradient ascent on the variable u.
- Gradient descent on the variable z.

Approximation of $\nabla_z F(z)$

Robbins-Monro algorithm to approximate u_z^* with U_z. We get a stochastic approximation of $\nabla_z F(z)$ with:

$$\hat{\omega}(z) = (\Gamma J_\sigma(z))^T U_z$$

Descent-Ascent Algorithm

- Gradient ascent on the variable u.
- Gradient descent on the variable z.

Double loop algorithm.

Addition of a regularizing term on h (M.Ballu et al. (2020))

$$\text{Regularization:} \qquad \varphi(h) = \sum_{k=1}^{K} h_k \log(h_k). \tag{11}$$

Addition of a regularizing term on h (M.Ballu et al. (2020))

$$\text{Regularization:} \qquad \varphi(h) = \sum_{k=1}^{K} h_k \log(h_k). \tag{11}$$

New optimization problem:

$$\min_{h \in \Sigma_k} W^{\varepsilon}(\alpha(h), \beta) + \lambda \varphi(h)$$

Addition of a regularizing term on h (M.Ballu et al. (2020))

$$\text{Regularization:} \qquad \varphi(h) = \sum_{k=1}^{K} h_k \log(h_k). \tag{11}$$

New optimization problem:

$$\min_{h \in \Sigma_k} W^{\varepsilon}(\alpha(h), \beta) + \lambda \varphi(h) = \min_{h \in \Sigma_k} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_\varepsilon(Y, u, h)] + \lambda \varphi(h)$$

Addition of a regularizing term on h (M.Ballu et al. (2020))

$$\text{Regularization:} \qquad \varphi(h) = \sum_{k=1}^{K} h_k \log(h_k). \tag{11}$$

New optimization problem:

$$\begin{aligned} \min_{h \in \Sigma_k} W^{\varepsilon}(\alpha(h), \beta) + \lambda\varphi(h) &= \min_{h \in \Sigma_k} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_\varepsilon(Y, u, h)] + \lambda\varphi(h) \\ &= \max_{u \in \mathbb{R}^I} \min_{h \in \Sigma_k} \mathbb{E}[g_\varepsilon(Y, u, h)] + \lambda\varphi(h) \end{aligned} \tag{12}$$

Addition of a regularizing term on h (M.Ballu et al. (2020))

$$\text{Regularization:} \qquad \varphi(h) = \sum_{k=1}^{K} h_k \log(h_k). \tag{11}$$

New optimization problem:

$$\begin{aligned} \min_{h \in \Sigma_k} W^{\varepsilon}(\alpha(h), \beta) + \lambda\varphi(h) &= \min_{h \in \Sigma_k} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, h)] + \lambda\varphi(h) \\ &= \max_{u \in \mathbb{R}^I} \min_{h \in \Sigma_k} \mathbb{E}[g_{\varepsilon}(Y, u, h)] + \lambda\varphi(h) \end{aligned} \tag{12}$$

for $u \in \mathbb{R}^I$, we can compute an explicit solution $h(u) \in \Sigma_K$ of the problem $\min_{h \in \Sigma_K} \mathbb{E}[g_{\varepsilon}(Y, u, h)] + \lambda\varphi(h)$.

$$k \in \{1, ..., K\}, \ (h(u))_k = \frac{\exp\left(-\frac{(\Gamma^T u)_k}{\lambda}\right)}{\sum_{l=1}^{K} \exp\left(-\frac{(\Gamma^T u)_l}{\lambda}\right)}$$

Using the expression of $h(u)$, problem (12) boils down to

$$\max_{u \in \mathbb{R}^I} \mathbb{E}_{Y \sim \beta}[f_{\varepsilon,\lambda}(Y, u)] \tag{13}$$

where Y is a random variable with distribution β.

Using the expression of $h(u)$, problem (12) boils down to

$$\max_{u \in \mathbb{R}^I} \mathbb{E}_{Y \sim \beta}[f_{\varepsilon,\lambda}(Y, u)] \tag{13}$$

where Y is a random variable with distribution β.

- $f_{\varepsilon,\lambda}(y_j, u)$ is easy to compute for $y_j \in \mathbb{R}^d$ an observation of Y, and $u \in \mathbb{R}^I$,

Using the expression of $h(u)$, problem (12) boils down to

$$\max_{u \in \mathbb{R}^I} \mathbb{E}_{Y \sim \beta}[f_{\varepsilon,\lambda}(Y, u)] \tag{13}$$

where Y is a random variable with distribution β.

- $f_{\varepsilon,\lambda}(y_j, u)$ is easy to compute for $y_j \in \mathbb{R}^d$ an observation of Y, and $u \in \mathbb{R}^I$,
- Estimate $\widehat{U}$ of a maximizer u^* of problem (15) with the Robbins-Monro algorithm,

Using the expression of $h(u)$, problem (12) boils down to

$$\max_{u \in \mathbb{R}^I} \mathbb{E}_{Y \sim \beta}[f_{\varepsilon,\lambda}(Y, u)] \tag{13}$$

where Y is a random variable with distribution β.

- $f_{\varepsilon,\lambda}(y_j, u)$ is easy to compute for $y_j \in \mathbb{R}^d$ an observation of Y, and $u \in \mathbb{R}^I$,
- Estimate $\widehat{U}$ of a maximizer u^* of problem (15) with the Robbins-Monro algorithm,
- From $\widehat{U}$ we derive an estimate of the class proportions $\hat{\pi} = h(\widehat{U})$.

Using the expression of $h(u)$, problem (12) boils down to

$$\max_{u \in \mathbb{R}^I} \mathbb{E}_{Y \sim \beta}[f_{\varepsilon,\lambda}(Y, u)] \tag{13}$$

where Y is a random variable with distribution β.

- $f_{\varepsilon,\lambda}(y_j, u)$ is easy to compute for $y_j \in \mathbb{R}^d$ an observation of Y, and $u \in \mathbb{R}^I$,
- Estimate $\widehat{U}$ of a maximizer u^* of problem (15) with the Robbins-Monro algorithm,
- From $\widehat{U}$ we derive an estimate of the class proportions $\hat{\pi} = h(\widehat{U})$.

Single loop algorithm.

Simulation study

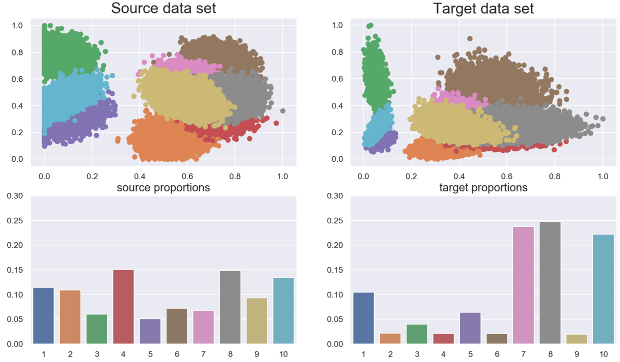


Figure 4: 2D projection of simulated data where $X_i \in \mathbb{R}^{10}$.

Spatial shift between the two data sets

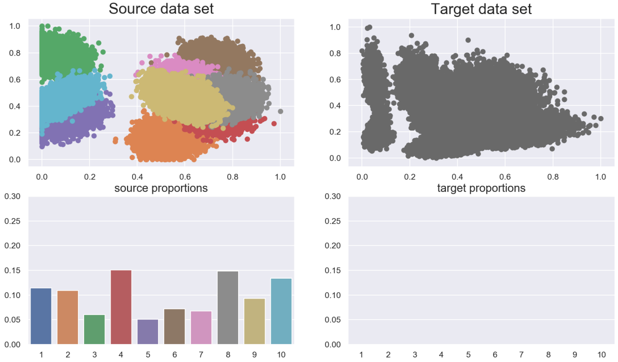
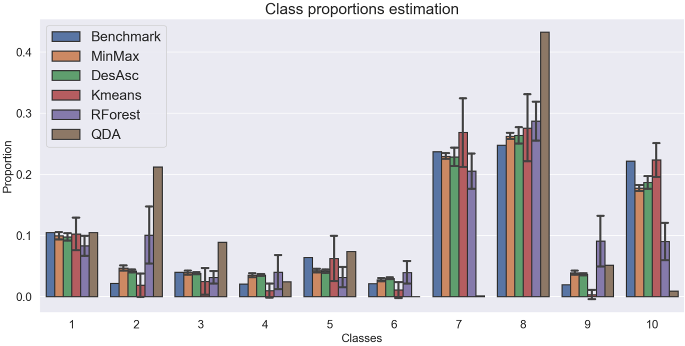


Figure 5: 2D projection of simulated data where $X_i \in \mathbb{R}^{10}$.

- Unsupervised method : Kmeans.
- Supervised methods : QDA and Random Forest.



Results on the flow cytometry data

Figure 6: Comparison of the estimated proportions $\hat{\pi}$ by `CytOpT` with the manual gating benchmark π. Source data set: Stanford1A.

Results on the flow cytometry data

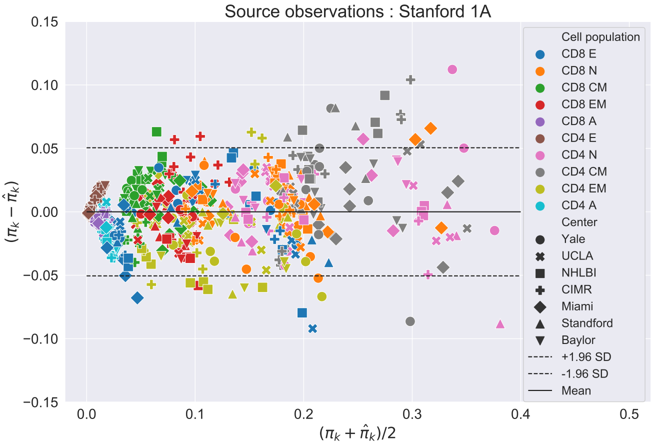


Figure 7: Comparison of the proportions $\hat{\pi}$ estimated with `CytOpT` and the manual benchmark π on the HIPC database.

Perspectives

$\alpha \in \mathcal{M}_+^1(\mathbb{R}^d)$ a probability measure that can be decomposed as a mixture of K probability measure $\alpha_1, ..., \alpha_K$:

$$\alpha = \sum_{k=1}^{K} \rho_k \alpha_k$$

.

Perspectives

$\alpha \in \mathcal{M}_+^1(\mathbb{R}^d)$ a probability measure that can be decomposed as a mixture of K probability measure $\alpha_1, ..., \alpha_K$:

$$\alpha = \sum_{k=1}^{K} \rho_k \alpha_k$$

. For $\theta \in \Sigma_K$ we define

$$\alpha_\theta = \sum_{k=1}^{K} \theta_k \alpha_k$$

.

Perspectives

$\alpha \in \mathcal{M}_+^1(\mathbb{R}^d)$ a probability measure that can be decomposed as a mixture of K probability measure $\alpha_1, ..., \alpha_K$:

$$\alpha = \sum_{k=1}^{K} \rho_k \alpha_k$$

. For $\theta \in \Sigma_K$ we define

$$\alpha_\theta = \sum_{k=1}^{K} \theta_k \alpha_k$$

.

Let $\beta \in \mathcal{M}_+^1(\mathbb{R}^d)$ an other probability measure. we define

$$\theta_* \in \arg\min_{\theta \in \Sigma_K} W(\alpha_\theta, \beta)$$

Empirical versions of α and β

- $\hat{\alpha} = \frac{1}{I}\sum_{i=1}^{I}\delta_{X_i} = \sum_{k=1}^{K}\frac{n_k}{I}\sum_{X_i\in C_k}\delta_{X_i} = \sum_{k=1}^{K}\frac{n_k}{I}\hat{\alpha}_k$.
- $\hat{\alpha}(h) = \sum_{k=1}^{K} h_k\hat{\alpha}_k$
- $\hat{\beta} = \frac{1}{J}\sum_{j=1}^{J}\delta_{Y_j}$

Empirical versions of α and β

- $\hat{\alpha} = \frac{1}{I}\sum_{i=1}^{I} \delta_{X_i} = \sum_{k=1}^{K} \frac{n_k}{I} \sum_{X_i \in C_k} \delta_{X_i} = \sum_{k=1}^{K} \frac{n_k}{I} \hat{\alpha}_k$.
- $\hat{\alpha}(h) = \sum_{k=1}^{K} h_k \hat{\alpha}_k$
- $\hat{\beta} = \frac{1}{J}\sum_{j=1}^{J} \delta_{Y_j}$

We denote

$$\theta_{\varepsilon,\lambda} \in \underset{h \in \Sigma_k}{\arg\min}\, W^{\varepsilon}(\hat{\alpha}(h), \hat{\beta}) + \lambda \varphi(h) \tag{14}$$

Empirical versions of α and β

- $\hat{\alpha} = \frac{1}{I}\sum_{i=1}^{I}\delta_{X_i} = \sum_{k=1}^{K}\frac{n_k}{I}\sum_{X_i\in C_k}\delta_{X_i} = \sum_{k=1}^{K}\frac{n_k}{I}\hat{\alpha}_k$.
- $\hat{\alpha}(h) = \sum_{k=1}^{K} h_k\hat{\alpha}_k$
- $\hat{\beta} = \frac{1}{J}\sum_{j=1}^{J}\delta_{Y_j}$

We denote

$$\theta_{\varepsilon,\lambda} \in \underset{h\in\Sigma_k}{\arg\min}\, W^{\varepsilon}(\hat{\alpha}(h),\hat{\beta}) + \lambda\varphi(h) \tag{14}$$

Question 1: How to choose ε, λ in order to minimize the quantity $\mathbb{E}\left[||\theta_{\varepsilon,\lambda} - \theta_*||^2\right]$?

Empirical versions of α and β

- $\hat{\alpha} = \frac{1}{I}\sum_{i=1}^{I}\delta_{X_i} = \sum_{k=1}^{K}\frac{n_k}{I}\sum_{X_i\in C_k}\delta_{X_i} = \sum_{k=1}^{K}\frac{n_k}{I}\hat{\alpha}_k.$
- $\hat{\alpha}(h) = \sum_{k=1}^{K} h_k\hat{\alpha}_k$
- $\hat{\beta} = \frac{1}{J}\sum_{j=1}^{J}\delta_{Y_j}$

We denote

$$\theta_{\varepsilon,\lambda} \in \underset{h\in\Sigma_k}{\arg\min}\, W^{\varepsilon}(\hat{\alpha}(h),\hat{\beta}) + \lambda\varphi(h) \tag{14}$$

Question 1: How to choose ε, λ in order to minimize the quantity $\mathbb{E}\left[||\theta_{\varepsilon,\lambda} - \theta_*||^2\right]$?
Question 2: Study of the convergence of the optimization algorithms to solve (14).

References

B.Bercu and J.Bigot (2020)

Asymptotic distribution and convergence rates of stochastic algorithms for entropic optimal transportation between probability measures

Annals of Statistics

I.Redko, N.Courty, R.Flamary and D.Tuia (2019)

Optimal Transport for Multi-source Domain Adaptation under Target Shift

International Conference on Artificial Intelligence and Statistics

G. Peyré et M.Cuturi (2018)

Computational Optimal Transport

Back up

$$
\begin{aligned}
f_{\varepsilon,\lambda}(y_j, u) = \varepsilon \left(\log(b_j) - \log\left(\sum_{i=1}^{I} \exp\left(\frac{u_i - c(x_i, y_j)}{\varepsilon} \right) \right) \right) \\
- \lambda \log\left(\sum_{l=1}^{K} \exp\left(-\frac{(\Gamma^T u)_l}{\lambda} \right) \right) - \varepsilon
\end{aligned}
\tag{15}
$$