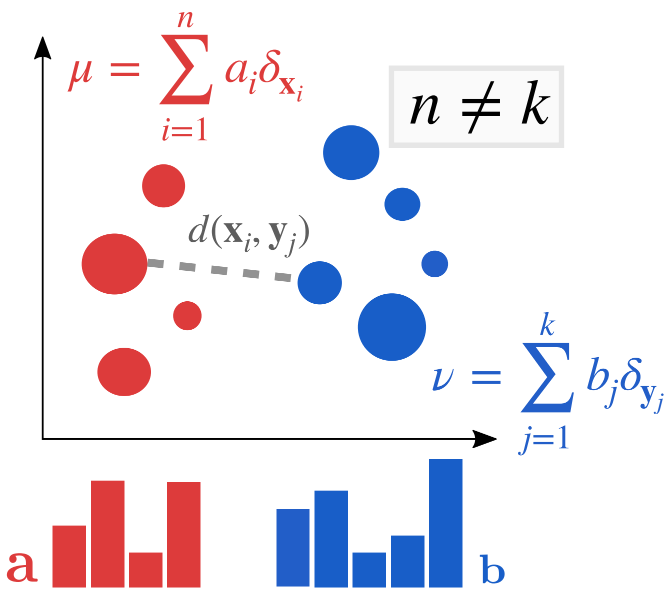
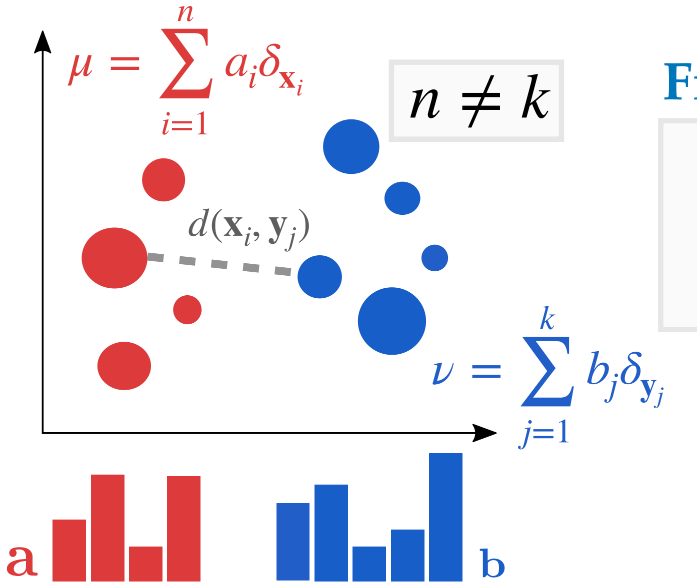
From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)



From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)

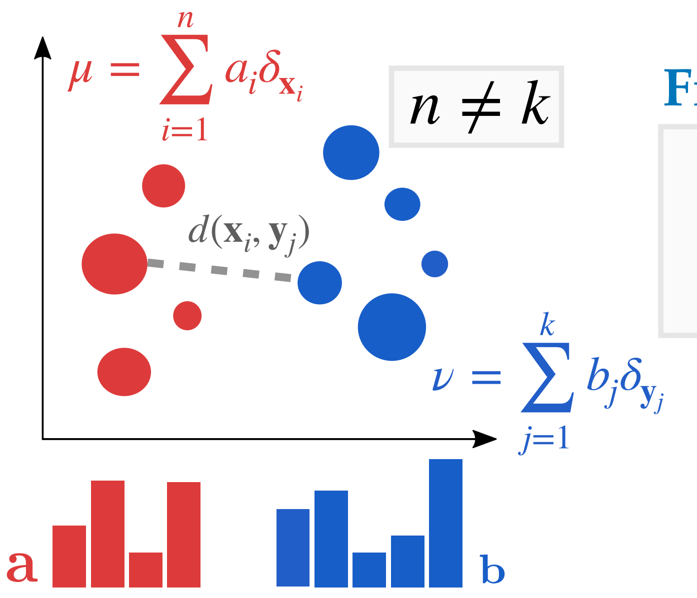


Find plan $T \in \mathbb{R}_+^{n \times k}$

$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij}$$

From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)



$n \neq k$

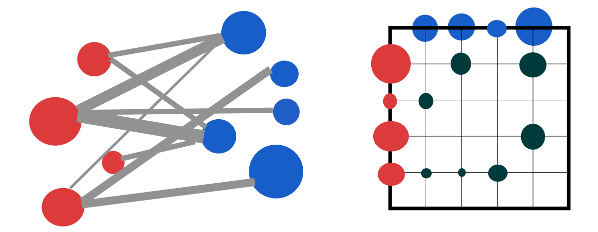
Find plan $T \in \mathbb{R}_+^{n \times k}$

$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij}$$

which constraints ?

Coupling $\Pi(a, b)$

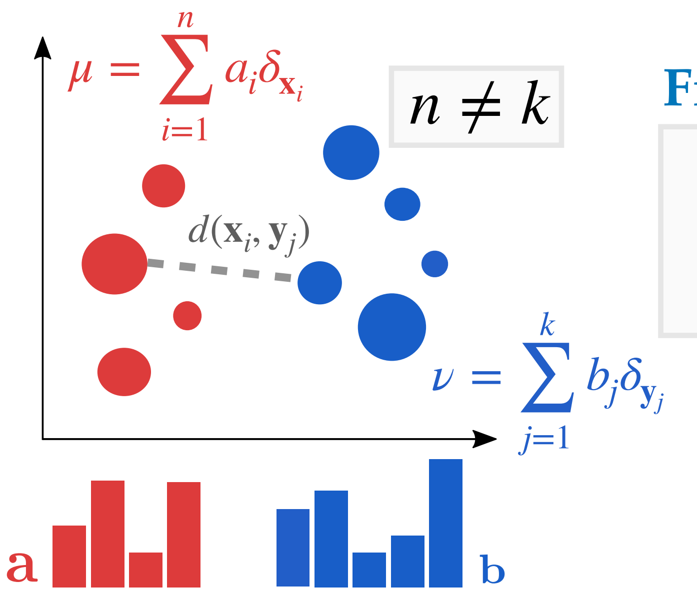
$$T^\top 1_n = a$$
$$T 1_k = b$$



$T \in [0,1]^{n \times k}$

From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)



$n \neq k$

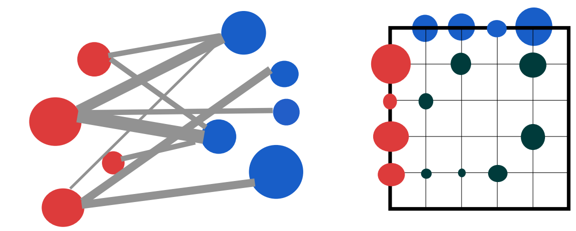
Find plan $T \in \mathbb{R}_+^{n \times k}$

$$\min_{T} \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij}$$

which constraints ?

Coupling $\Pi(a, b)$

$$T^\top 1_n = a$$
$$T 1_k = b$$



$T \in [0,1]^{n \times k}$

✦ Wasserstein distance

$\mu \in \mathscr{P}(X)$

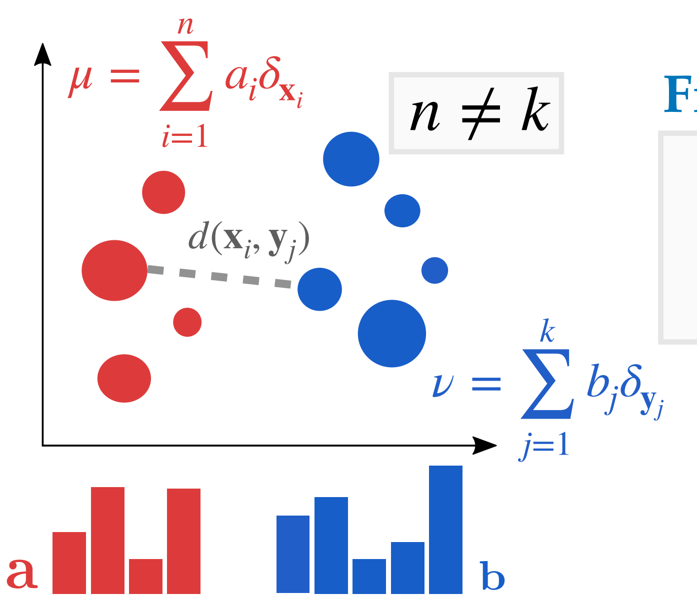
$\nu \in \mathscr{P}(X)$

$$\mathrm{W}_p(\mu, \nu) = \left(\min_{T} \int_{X \times X} d(x, y)^p \, \mathrm{d}T(x, y) \right)^{1/p}$$

- ✦ It is always **well-defined**
- ✦ It is a proper distance on $\mathscr{P}(X)$
- ✦ Lifts the geometry of $X \rightarrow \mathscr{P}(X)$

From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)



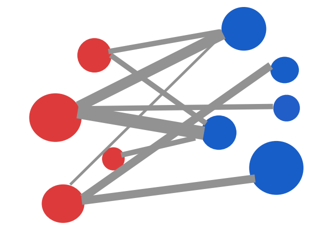
Find plan $T \in \mathbb{R}_+^{n \times k}$

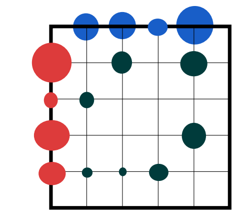
$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij}$$

which constraints ?

Coupling $\Pi(a, b)$

$$T^\top 1_n = a$$
$$T 1_k = b$$





$T \in [0,1]^{n \times k}$

✦ Wasserstein distance

$\mu \in \mathscr{P}(X)$

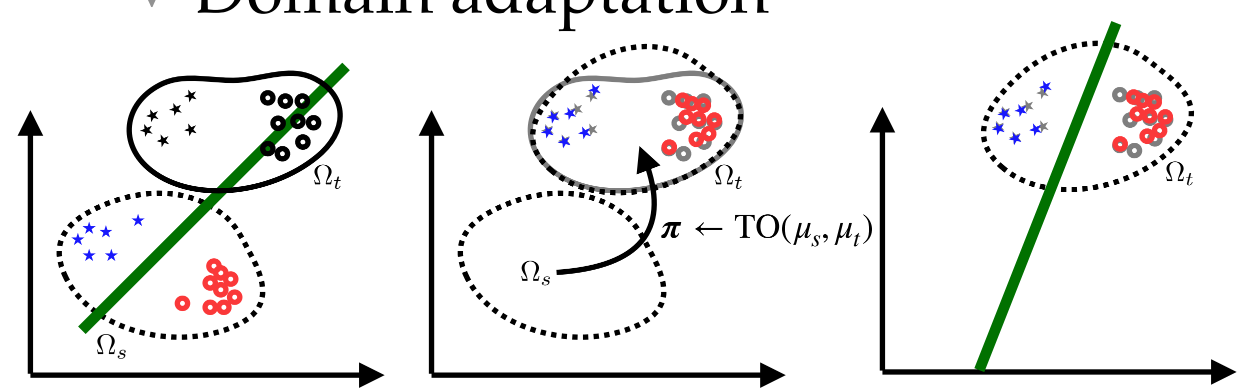
$\nu \in \mathscr{P}(X)$

$$W_p(\mu, \nu) = \left(\min_T \int_{X \times X} d(x, y)^p \, \mathrm{d}T(x, y) \right)^{1/p}$$

- It is always **well-defined**
- It is a proper distance on $\mathscr{P}(X)$
- Lifts the geometry of $X \rightarrow \mathscr{P}(X)$

✦ In machine learning

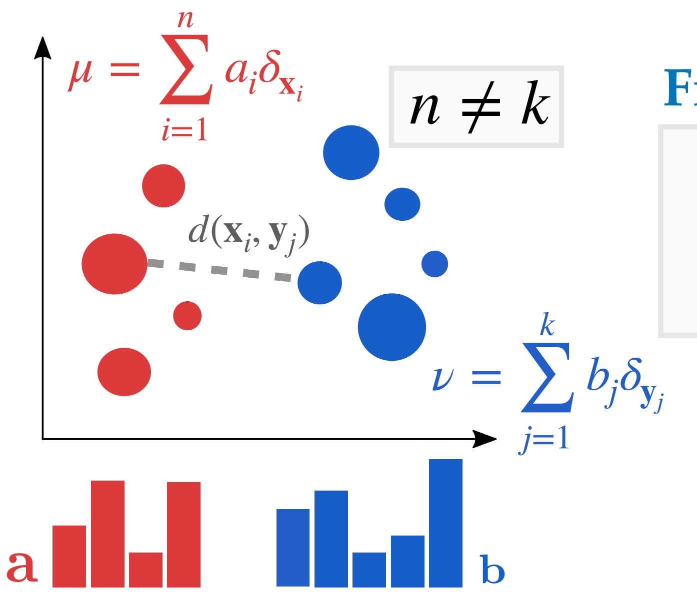
- Domain adaptation



- Generative modeling
- Analysis of NN convergence
- ML on graphs, fairness
- And many other …

From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)



$n \neq k$

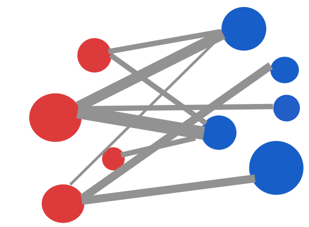
Find plan $T \in \mathbb{R}_+^{n \times k}$

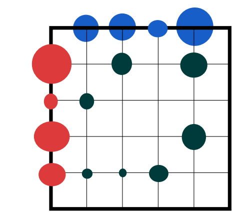
$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij}$$

which constraints ?

Coupling $\Pi(a, b)$

$$T^\top 1_n = a$$
$$T 1_k = b$$





$T \in [0,1]^{n \times k}$

✦ Algorithmic fundations

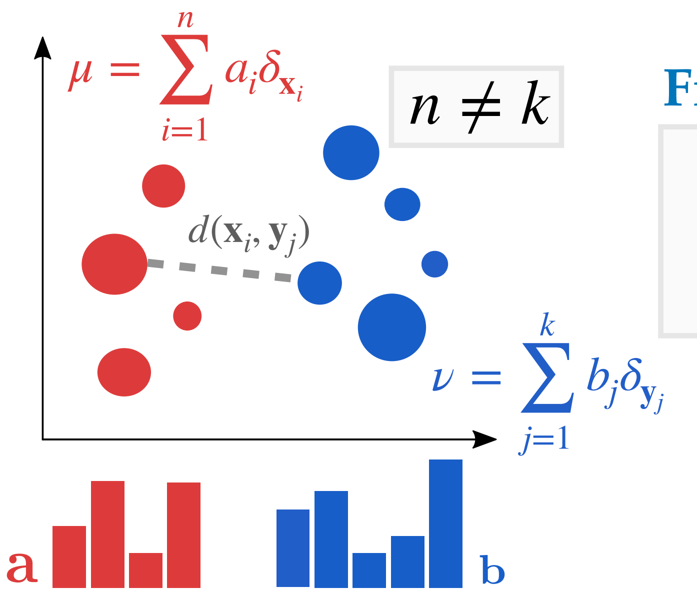
Unregularized problem

✦ Simplex, Network flow

$$\mathcal{O}(n^3 \log(n)^2)$$

From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)



$n \neq k$

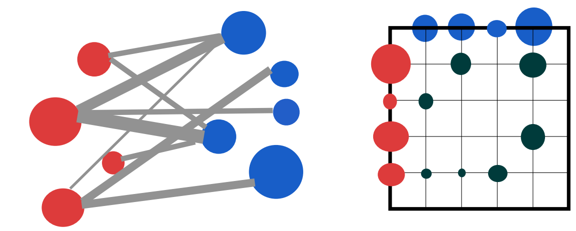
Find plan $T \in \mathbb{R}_+^{n \times k}$

$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij}$$

which constraints ?

Coupling $\Pi(a, b)$

$$T^\top 1_n = a$$
$$T 1_k = b$$



$T \in [0,1]^{n \times k}$

✦ Algorithmic fundations

Unregularized problem

✦ Simplex, Network flow

$$\mathcal{O}(n^3 \log(n)^2)$$

Entropic regularization

$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij} - \varepsilon H(T)$$

Sinkhorn algorithm

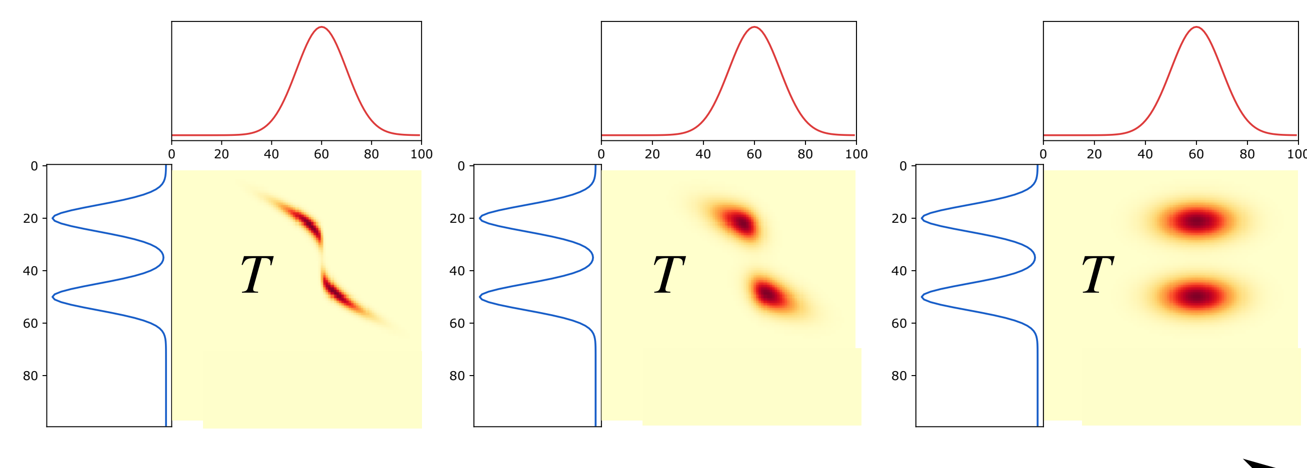
$$K := \left(e^{-d(\mathbf{x}_i, \mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$
$$v = b \oslash K u$$

```
output
```

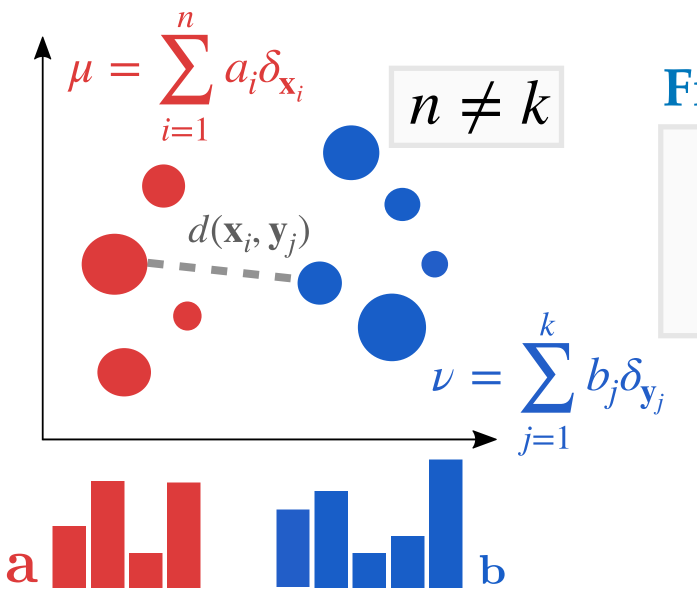
$$T = \mathrm{diag}(u) K \, \mathrm{diag}(v)$$



$\epsilon \to +\infty$

From Wasserstein to Sinkhorn

✦ Classical optimal transport (in a nutshell)



$n \neq k$

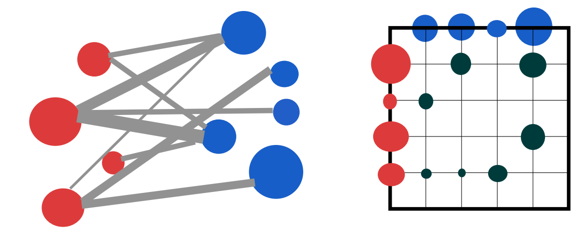
Find plan $T \in \mathbb{R}_+^{n \times k}$

$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij}$$

which constraints ?

Coupling $\Pi(a, b)$

$$T^\top \mathbb{1}_n = a$$
$$T \mathbb{1}_k = b$$



$T \in [0,1]^{n \times k}$

✦ Algorithmic fundations

Unregularized problem

✦ Simplex, Network flow

$$\mathcal{O}(n^3 \log(n)^2)$$

Regularized problem

$$\mathcal{O}(n^2)$$

Entropic regularization

$$\min_T \sum_{ij} d(\mathbf{x}_i, \mathbf{y}_j) T_{ij} - \varepsilon H(T)$$

Sinkhorn algorithm

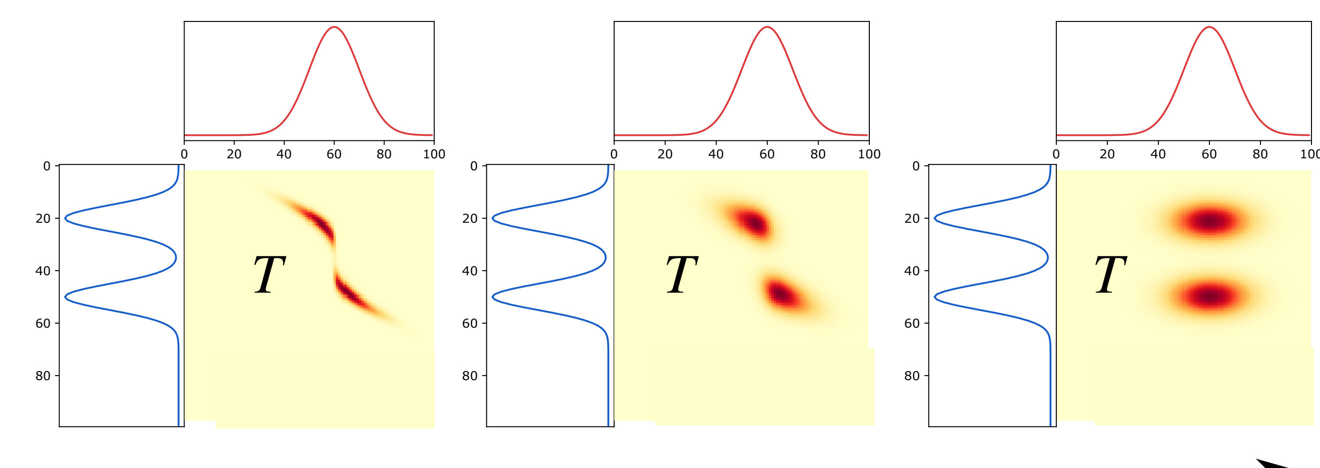
$$K := \left(e^{-d(\mathbf{x}_i, \mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$
$$v = b \oslash K u$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$



$\epsilon \to +\infty$

Accelerating Sinkhorn algorithm

✦ **Goal: fast approximation of** $u \to Ku$

Sinkhorn algorithm

$$K := \left(e^{-d(\mathbf{x}_i,\mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$
$$v = b \oslash Ku$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$

Accelerating Sinkhorn algorithm

✦ **Goal: fast approximation of** $u \to Ku$

Sinkhorn algorithm

$$K := \left(e^{-d(\mathbf{x}_i, \mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

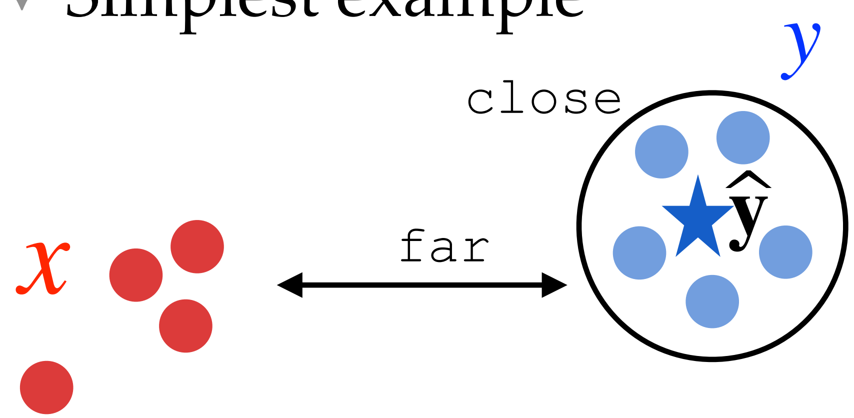
$$u = a \oslash K^\top v$$

$$v = b \oslash Ku$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$

✦ Simplest example



$$K \approx \mathbf{p}\mathbf{q}^\top = \begin{pmatrix} k(\mathbf{x}_1, \hat{\mathbf{y}}) \\ k(\mathbf{x}_2, \hat{\mathbf{y}}) \\ \vdots \\ k(\mathbf{x}_n, \hat{\mathbf{y}}) \end{pmatrix} 1^\top$$

$u \to \tilde{K}u$ in $\mathcal{O}(n)$

Accelerating Sinkhorn algorithm

✦ **Goal: fast approximation of** $u \to Ku$

Sinkhorn algorithm

$$K := \left(e^{-d(\mathbf{x}_i, \mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$

$$v = b \oslash Ku$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$

✦ Simplest example

$$K \approx \mathbf{p}\mathbf{q}^\top = \begin{pmatrix} k(\mathbf{x}_1, \hat{\mathbf{y}}) \\ k(\mathbf{x}_2, \hat{\mathbf{y}}) \\ \vdots \\ k(\mathbf{x}_n, \hat{\mathbf{y}}) \end{pmatrix} 1^\top$$

$$u \to \tilde{K}u \text{ in } \mathcal{O}(n)$$

✦ Generally if r clusters in y far enough from all the x

$$K \approx \mathbf{P}\mathbf{Q}^\top \quad u \to \tilde{K}u \text{ in } \mathcal{O}(rn)$$

✦ But unknown clusters + crude approximation !

Accelerating Sinkhorn algorithm

✦ **Goal: fast approximation of** $u \to Ku$

Sinkhorn algorithm

$$K := \left(e^{-d(\mathbf{x}_i,\mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$
$$v = b \oslash Ku$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$

✦ Simplest example

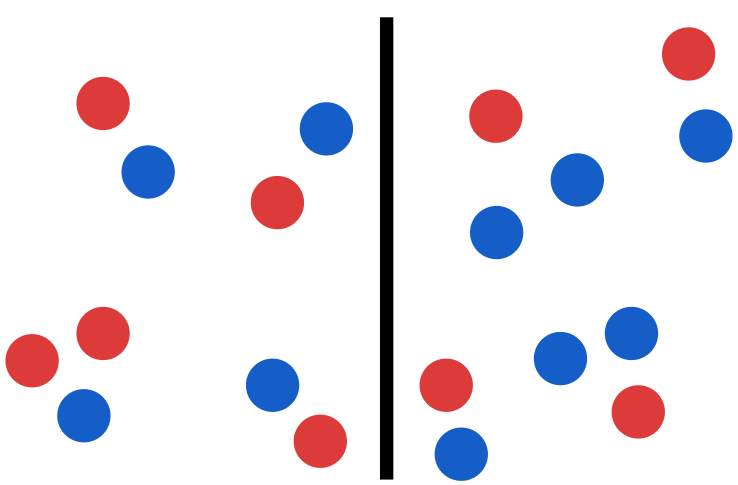
$$K \approx \mathbf{p}\mathbf{q}^\top = \begin{pmatrix} k(\mathbf{x}_1, \hat{\mathbf{y}}) \\ k(\mathbf{x}_2, \hat{\mathbf{y}}) \\ \vdots \\ k(\mathbf{x}_n, \hat{\mathbf{y}}) \end{pmatrix} 1^\top$$

$$u \to \tilde{K}u \text{ in } \mathcal{O}(n)$$

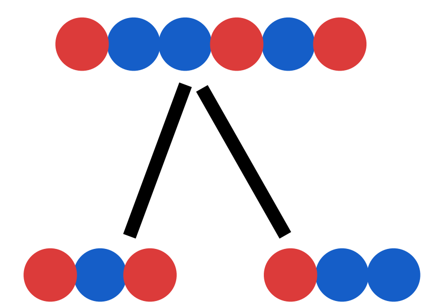
✦ Generally if r clusters in y far enough from all the x

$$K \approx \mathbf{P}\mathbf{Q}^\top \quad u \to \tilde{K}u \text{ in } \mathcal{O}(rn)$$

✦ **Idea: hierarchical clustering**



$$K = \left[\begin{array}{c|c} & \mathbf{x} \in C, \mathbf{y} \in \overline{C} \\ \hline \mathbf{x} \in \overline{C}, \mathbf{y} \in C & \end{array}\right]$$



Accelerating Sinkhorn algorithm

✦ **Goal: fast approximation of** $u \to Ku$

Sinkhorn algorithm

$$K := \left(e^{-d(\mathbf{x}_i,\mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$
$$v = b \oslash Ku$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$

✦ Simplest example

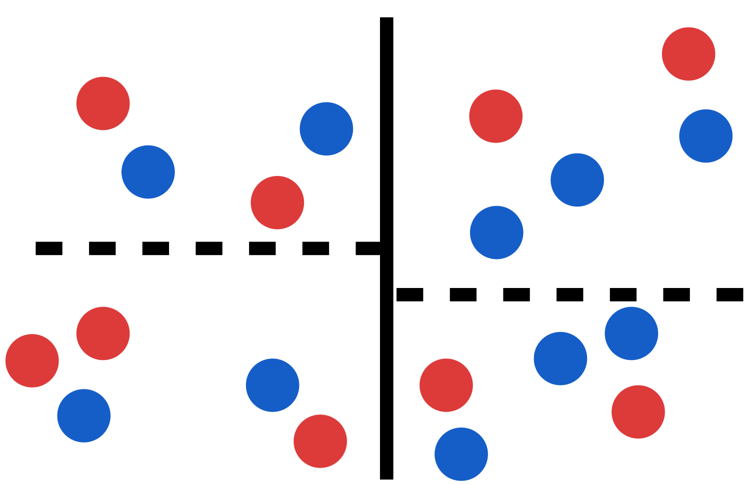
$$K \approx \mathbf{p}\mathbf{q}^\top = \begin{pmatrix} k(\mathbf{x}_1, \hat{\mathbf{y}}) \\ k(\mathbf{x}_2, \hat{\mathbf{y}}) \\ \vdots \\ k(\mathbf{x}_n, \hat{\mathbf{y}}) \end{pmatrix} 1^\top$$

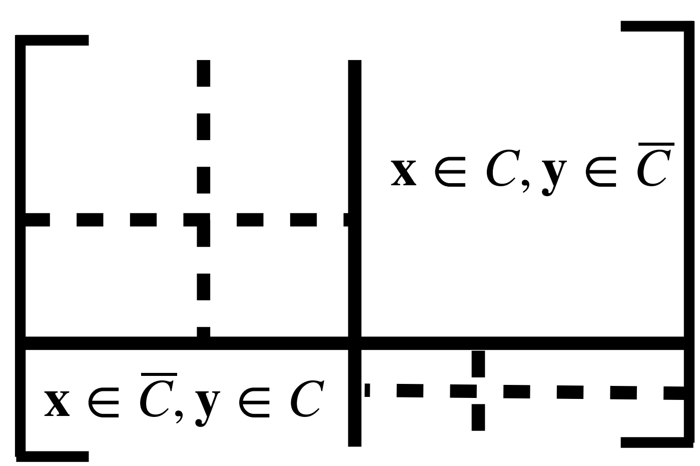
$$u \to \tilde{K}u \text{ in } \mathcal{O}(n)$$

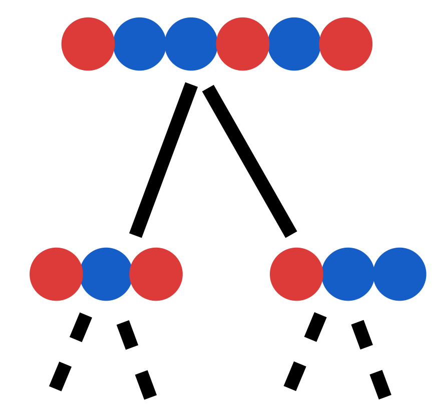
✦ Generally if r clusters in y far enough from all the x

$$K \approx \mathbf{P}\mathbf{Q}^\top \quad u \to \tilde{K}u \text{ in } \mathcal{O}(rn)$$

✦ **Idea: hierarchical clustering**



$K =$ 



Accelerating Sinkhorn algorithm

✦ **Goal: fast approximation of** $u \to Ku$

Sinkhorn algorithm

$$K := \left(e^{-d(\mathbf{x}_i,\mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$
$$v = b \oslash Ku$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$

✦ Simplest example

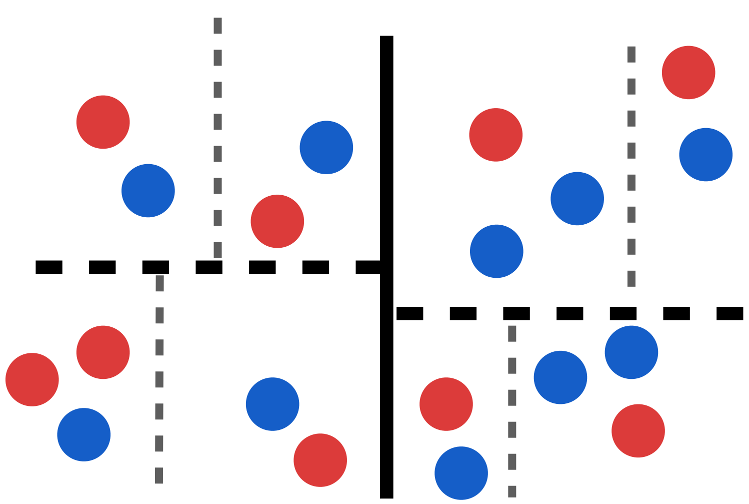
$$K \approx \mathbf{p}\mathbf{q}^\top = \begin{pmatrix} k(\mathbf{x}_1, \hat{\mathbf{y}}) \\ k(\mathbf{x}_2, \hat{\mathbf{y}}) \\ \vdots \\ k(\mathbf{x}_n, \hat{\mathbf{y}}) \end{pmatrix} 1^\top$$

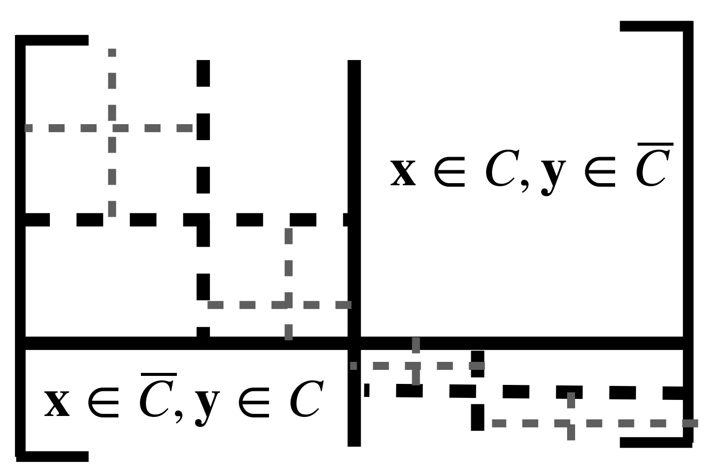
$$u \to \tilde{K}u \text{ in } \mathcal{O}(n)$$

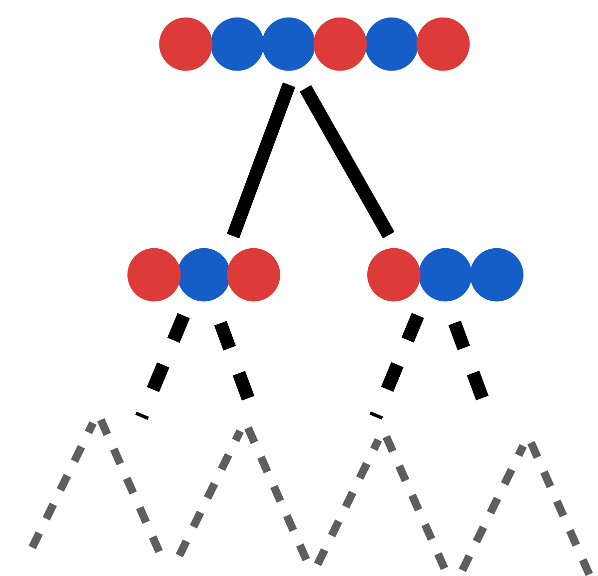
✦ Generally if r clusters in y far enough from all the x

$$K \approx \mathbf{P}\mathbf{Q}^\top \quad u \to \tilde{K}u \text{ in } \mathcal{O}(rn)$$

✦ **Idea: hierarchical clustering**



$K =$ 



Accelerating Sinkhorn algorithm

✦ **Goal: fast approximation of** $u \to Ku$

Sinkhorn algorithm

$$K := \left(e^{-d(\mathbf{x}_i, \mathbf{y}_j)/\varepsilon}\right)_{ij}$$

```
while not converged:
```

$$u = a \oslash K^\top v$$
$$v = b \oslash Ku$$

```
output
```

$$T = \mathrm{diag}(u) K \,\mathrm{diag}(v)$$

✦ Simplest example

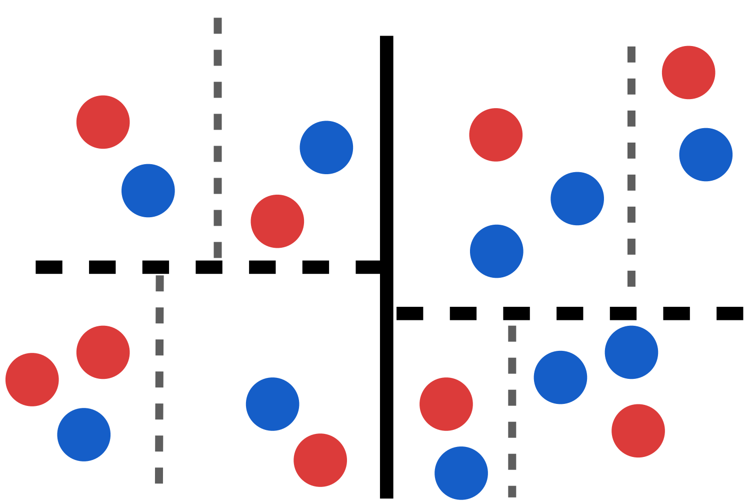
$$K \approx \mathbf{p}\mathbf{q}^\top = \begin{pmatrix} k(\mathbf{x}_1, \hat{\mathbf{y}}) \\ k(\mathbf{x}_2, \hat{\mathbf{y}}) \\ \vdots \\ k(\mathbf{x}_n, \hat{\mathbf{y}}) \end{pmatrix} 1^\top$$

$$u \to \tilde{K}u \text{ in } \mathcal{O}(n)$$

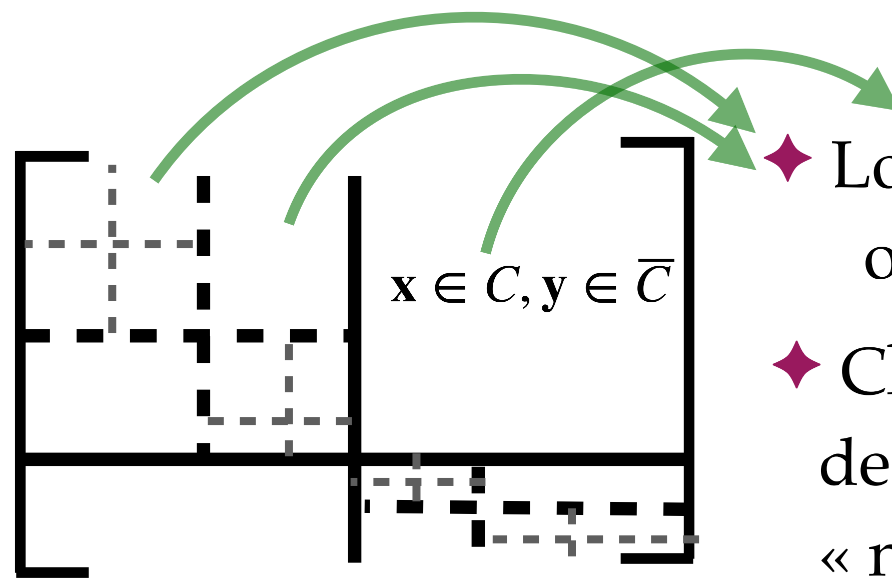
✦ Generally if r clusters in y far enough from all the x

$$K \approx \mathbf{P}\mathbf{Q}^\top \quad u \to \tilde{K}u \text{ in } \mathcal{O}(rn)$$

✦ **Idea: hierarchical clustering**



$K =$



✦ Low-rank approx of « far interactions »

✦ Cheap tree traversal to detect « far » and « near » interactions

✦ Fast multipole methods, Barnes-Hut algorithm, H-matrices