ICML - Kernels & RKHS Workshop

Distances and Kernels for Structured Objects

Marco Cuturi - Kyoto University

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When observations are in $\mathbb{R}^n$
  - **Distances** and **Positive Definite Kernels** share many properties

Outline

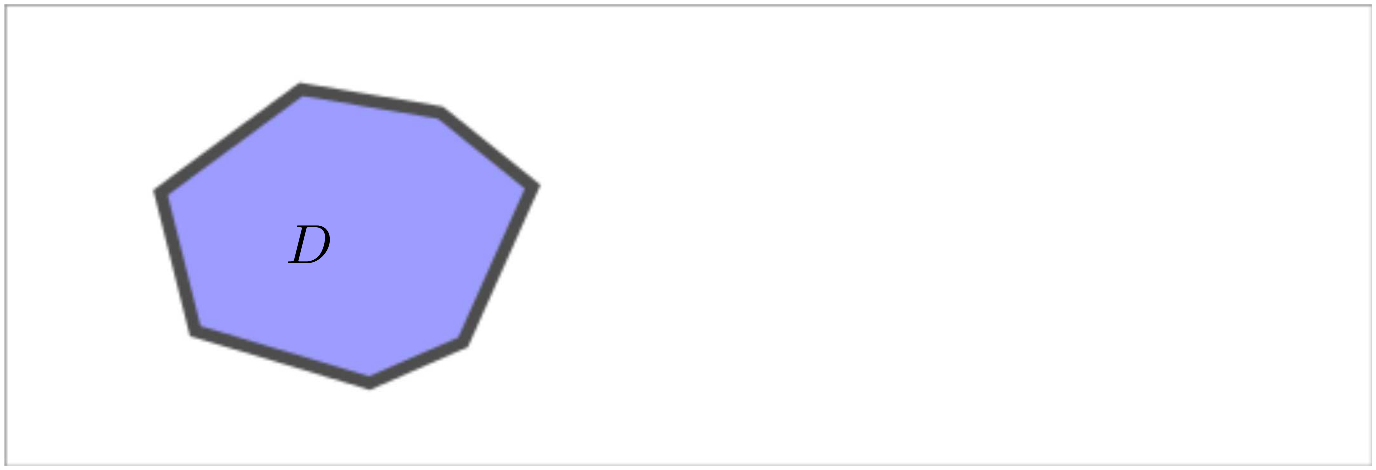
Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When observations are in $\mathbb{R}^n$
  - **Distances** and **Positive Definite Kernels** share many properties
  - At their interface lies the family of **Negative Definite Kernels**

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

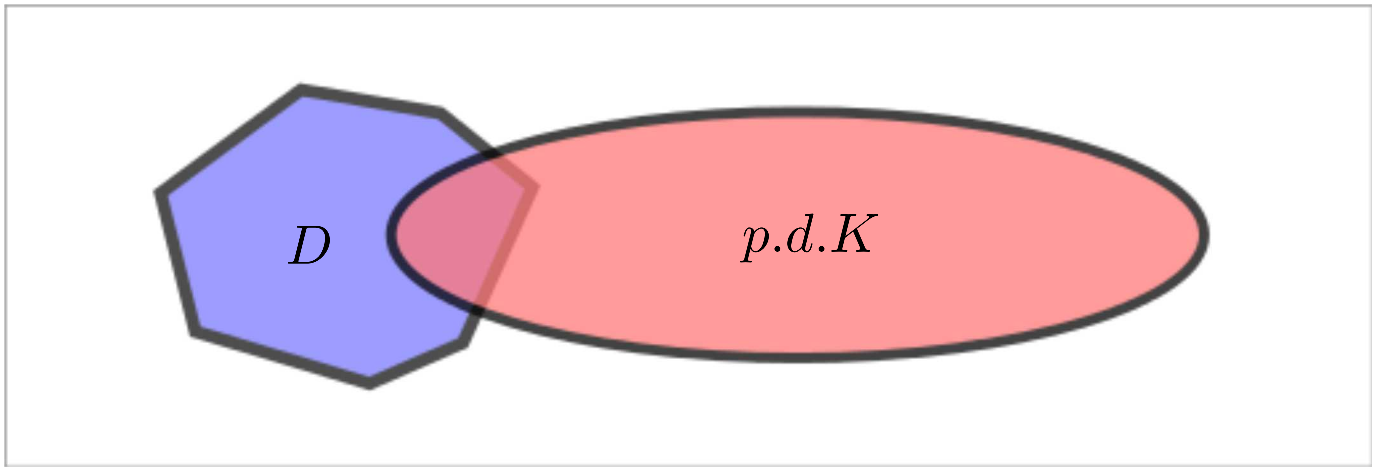
- When observations are in $\mathbb{R}^n$



Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When observations are in $\mathbb{R}^n$

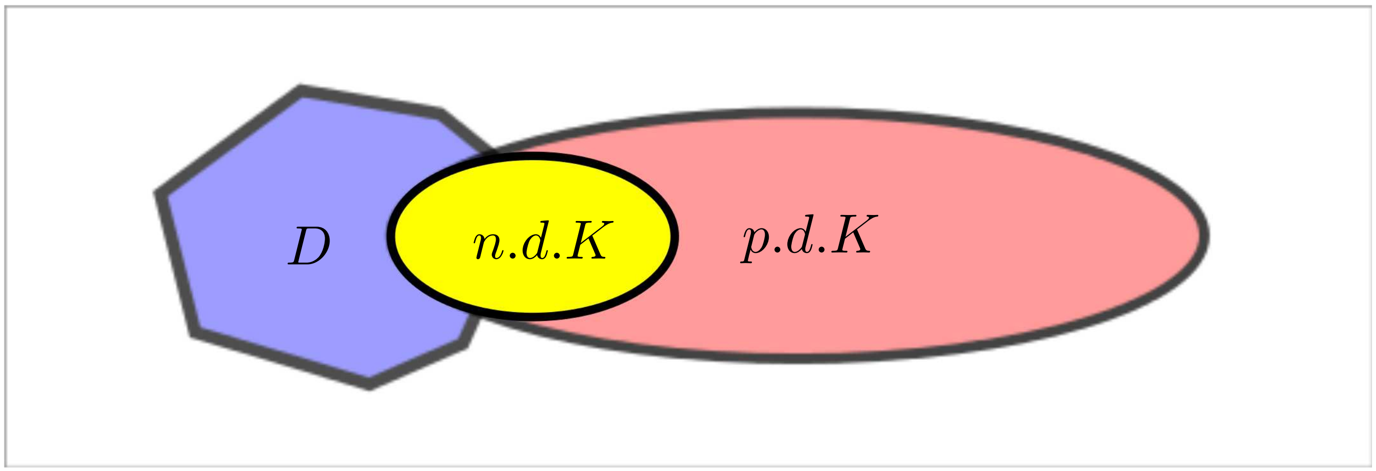


(note: intersection not to be taken literally)

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

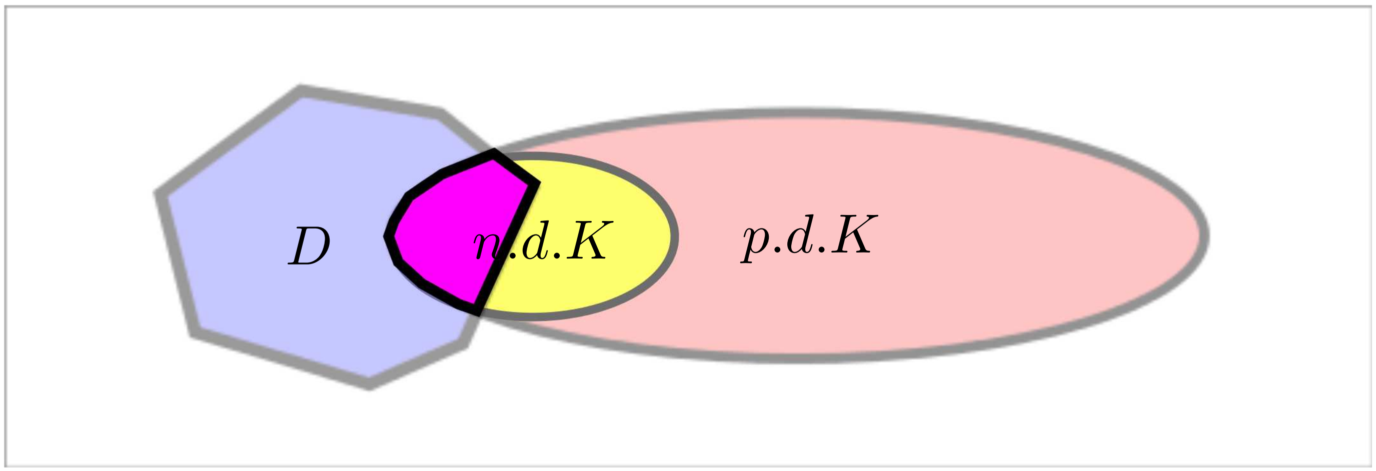
- When observations are in $\mathbb{R}^n$



Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When observations are in $\mathbb{R}^n$



- Hilbertian metrics are a sweet spot, both in theory and practice.

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When comparing **structured data** (constrained subsets of $\mathbb{R}^n$, n very large)...

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When comparing **structured data** (constrained subsets of $\mathbb{R}^n$, n very large)
  - **Classical distances** on $\mathbb{R}^n$ that ignore such constraints perform poorly

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When comparing **structured data** (constrained subsets of $\mathbb{R}^n$, n very large)
  - **Classical distances** on $\mathbb{R}^n$ that ignore such constraints perform poorly
  - **Combinatorial distances** (to be defined) take them into account
    (string, tree) Edit-distances, DTW, optimal matchings, transportation distances

Outline

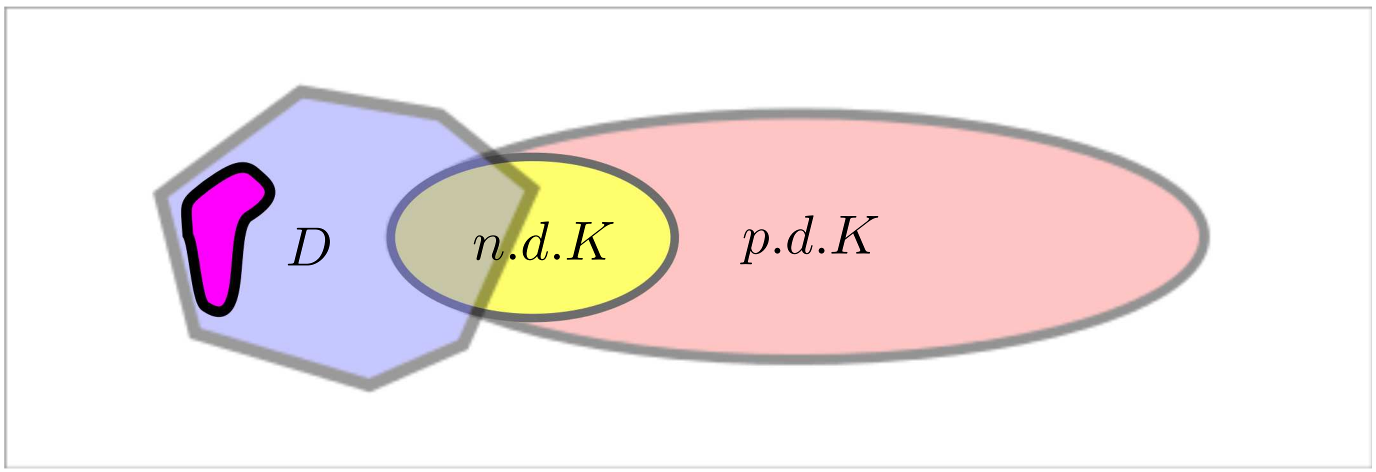
Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When comparing **structured data** (constrained subsets of $\mathbb{R}^n$, n very large)
  - **Classical distances** on $\mathbb{R}^n$ that ignore such constraints perform poorly
  - **Combinatorial distances** (to be defined) take them into account
    (string, tree) Edit-distances, DTW, optimal matchings, transportation distances
  - **Combinatorial distances are not negative definite** (in the general case)

Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

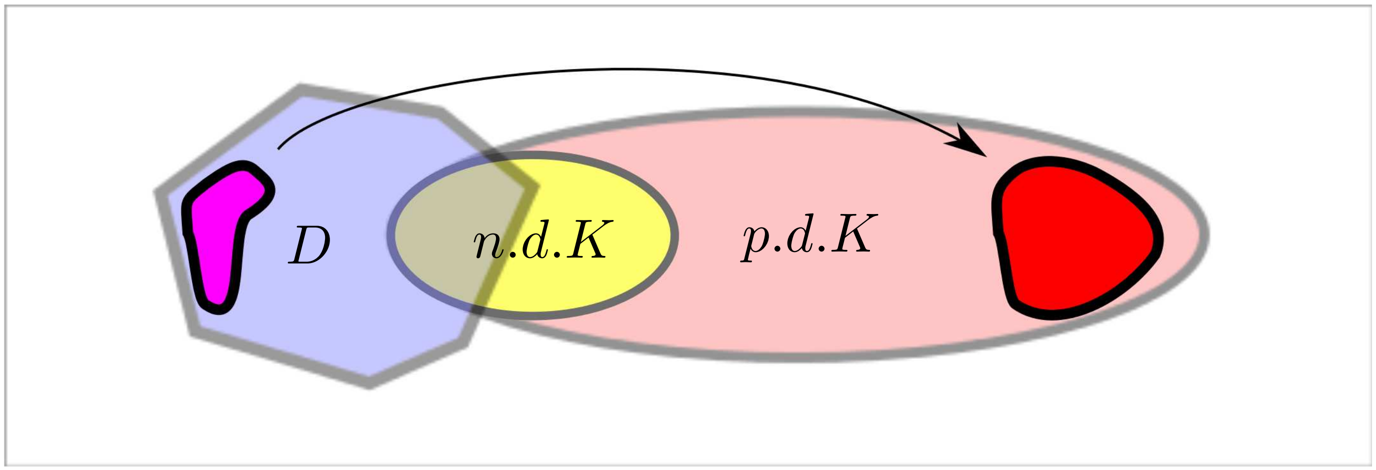
- When comparing **structured data** (constrained subsets of $\mathbb{R}^n$, n very large)



Outline

Distances and **Positive Definite Kernels** are crucial ingredients in many popular ML algorithms .

- When comparing **structured data** (constrained subsets of $\mathbb{R}^n$, n very large)



Main message of this talk:

we can recover p.d. kernels from combinatorial distances through generating functions.

Distances and Kernels

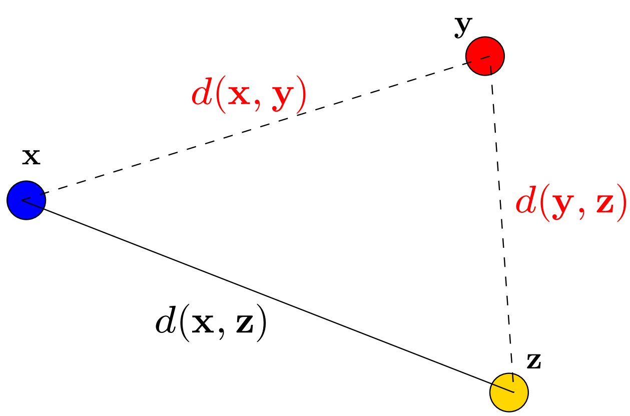
Distances

A bivariate function defined on a set $\mathcal{X}$,

$$\begin{array}{rccc} \boldsymbol{d}: & \mathcal{X} \times \mathcal{X} & \to & \mathbb{R}_+ \\ & (\mathbf{x}, \mathbf{y}) & \mapsto & \boldsymbol{d}(\mathbf{x}, \mathbf{y}) \end{array}$$

is a **distance** if $\forall\, \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$,

- $\boldsymbol{d}(\mathbf{x}, \mathbf{y}) = \boldsymbol{d}(\mathbf{y}, \mathbf{x})$, *symmetry*
- $\boldsymbol{d}(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$, *definiteness*
- $\boldsymbol{d}(\mathbf{x}, \mathbf{z}) \leq \boldsymbol{d}(\mathbf{x}, \mathbf{y}) + \boldsymbol{d}(\mathbf{y}, \mathbf{z})$, *triangle inequality*



Kernels (Symmetric & Positive Definite)

A bivariate function defined on a set $\mathcal{X}$

$$\begin{aligned} \boldsymbol{k}: \quad \mathcal{X} \times \mathcal{X} & \rightarrow \quad \mathbb{R}_+ \\ (\mathbf{x}, \mathbf{y}) & \mapsto \quad \boldsymbol{k}(\mathbf{x}, \mathbf{y}) \end{aligned}$$

is a **positive definite kernel** if $\forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X}$,

- $\boldsymbol{k}(\mathbf{x}, \mathbf{y}) = \boldsymbol{k}(\mathbf{y}, \mathbf{x})$, *symmetry*

and $\forall\, n \in \mathbb{N}, \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}^n, c \in \mathbb{R}^n$

- $\sum_{i=1}^{n} c_i\, c_j\, \boldsymbol{k}(\mathbf{x}_i, \mathbf{x}_j) \geq 0$

Matrices

$$\boxed{\begin{array}{c}\text{Convex cone of } n \times n \textbf{ distance} \text{ matrices - dimension } \frac{n(n-1)}{2} \\ \boldsymbol{\mathcal{M}_n} = \{X \in \mathbb{R}^{n \times n} \,|\, x_{ii} = 0; \text{ for } i < j, x_{ij} > 0; x_{i\boldsymbol{k}} + x_{\boldsymbol{k}j} - x_{ij} \geq 0\}\end{array}}$$

$3\binom{3}{n} + \binom{2}{n}$ linear inequalities; n equalities

Matrices

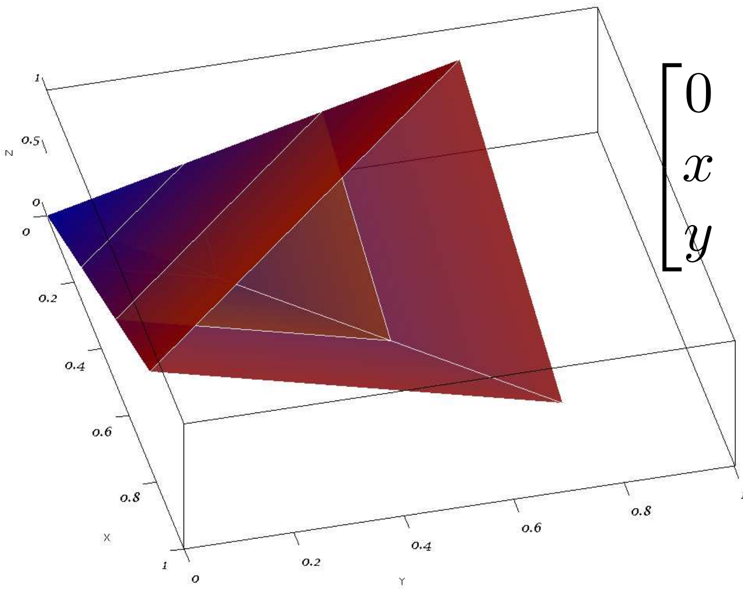
Convex cone of $n \times n$ **distance** matrices - dimension $\frac{n(n-1)}{2}$
$\boldsymbol{\mathcal{M}_n} = \{X \in \mathbb{R}^{n\times n} \,|\, x_{ii} = 0;\ \text{for } i \neq j, x_{ij} > 0; x_{i\boldsymbol{k}} + x_{\boldsymbol{k}j} - x_{ij} \geq 0\}$

$3\binom{3}{n} + \binom{2}{n}$ linear inequalities; n equalities

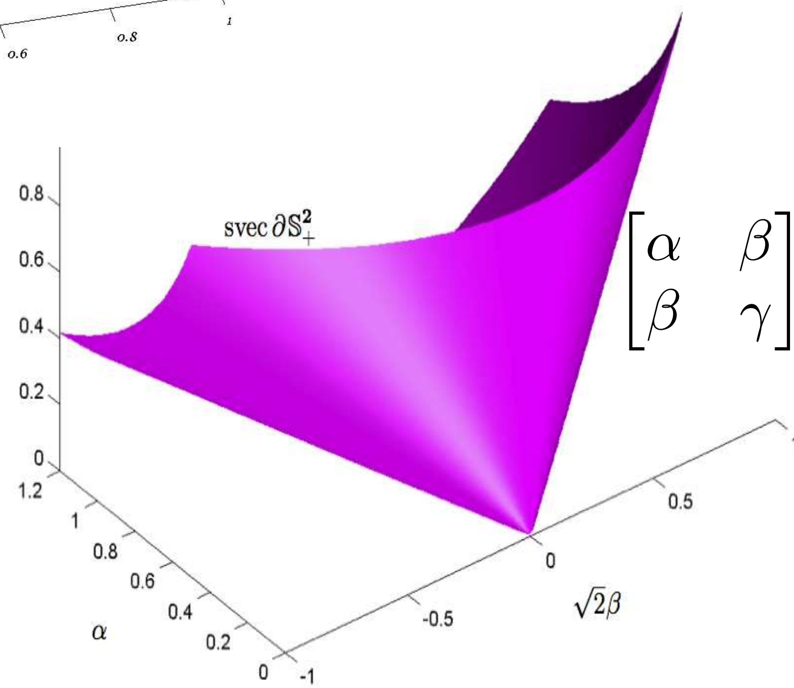
Convex cone of $n \times n$ **p.s.d. matrices** - dimension $\frac{n(n+1)}{2}$
$\boldsymbol{\mathcal{S}_n^+} = \{X \in \mathbb{R}^{n\times n} \,|\, X = X^T; \forall \mathbf{z} \in \mathbb{R}^n,\ \mathbf{z}^T X \mathbf{z} \geq 0\}$

$\forall \mathbf{z} \in \mathbb{R}^n, \langle X, \mathbf{z}\mathbf{z}^T \rangle \geq 0$: infinite number of inequalities; $\binom{2}{n}$ equalities

Cones



$$\begin{bmatrix} 0 & x & y \\ x & 0 & z \\ y & z & 0 \end{bmatrix}$$



$$\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}$$

∂S^2_+ image: Dattoro

Functions & Matrices

$$\boldsymbol{d} \text{ distance} \quad \Leftrightarrow \quad \forall\, n \in \mathbb{N}, \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}^n \quad \big[\boldsymbol{d}(\mathbf{x}_i, \mathbf{x}_j)\big] \in \boldsymbol{\mathcal{M}_n}$$

$$\boldsymbol{k} \text{ kernel} \quad \Leftrightarrow \quad \forall\, n \in \mathbb{N}, \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}^n \quad \big[\boldsymbol{k}(\mathbf{x}_i, \mathbf{x}_j)\big] \in \boldsymbol{\mathcal{S}_n^+}$$

Extreme Rays & Facets

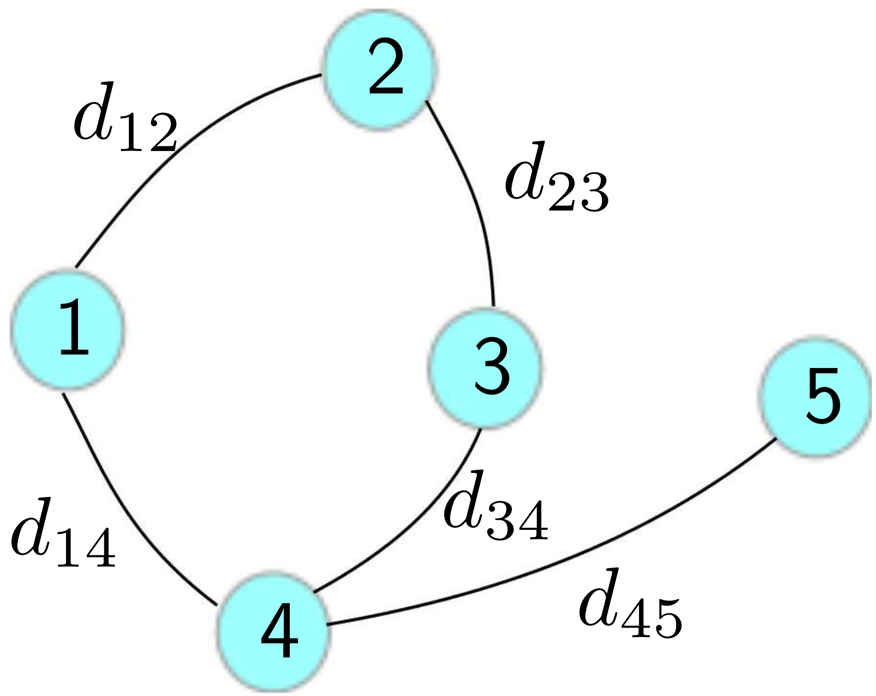
$\mathcal{M}_n$ is a polyhedral cone.

- Facets $= 3\binom{3}{n}$ hyperplanes $d_{ik} + d_{kj} - d_{ij} = 0$.
- Avis (1980) shows that extreme rays are arbitrarily complex using graph metrics

Extreme Rays & Facets

$\mathcal{M}_n$ is a polyhedral cone.

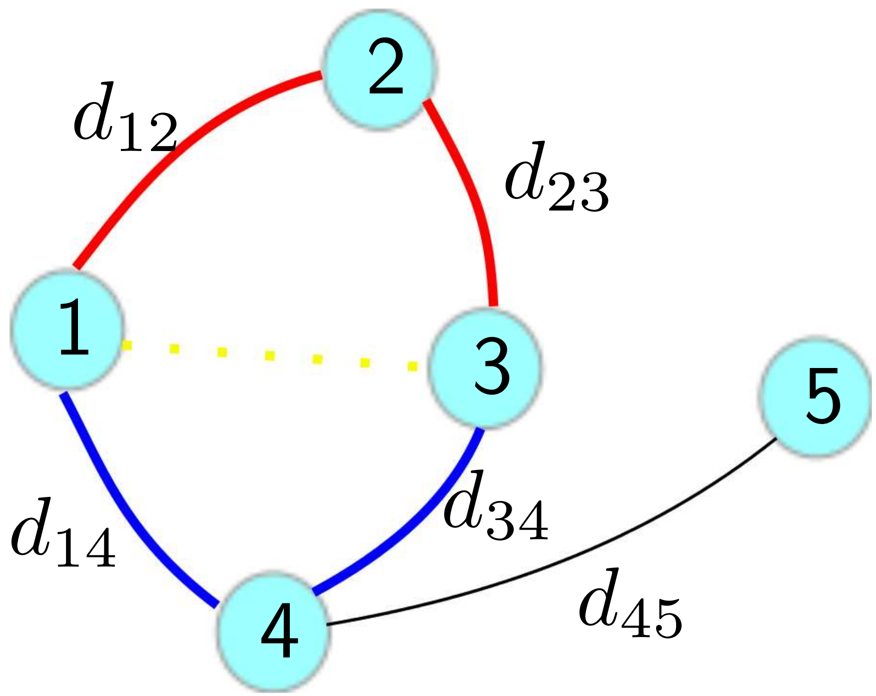
- Facets $= 3\binom{3}{n}$ hyperplanes $d_{ik} + d_{kj} - d_{ij} = 0$.
- Avis (1980) shows that extreme rays are arbitrarily complex using graph metrics



Extreme Rays & Facets

$\mathcal{M}_n$ is a polyhedral cone.

- Facets $= 3\binom{3}{n}$ hyperplanes $d_{ik} + d_{kj} - d_{ij} = 0$.
- Avis (1980) shows that extreme rays are arbitrarily complex using graph metrics



$$d_{13} = \min(d_{12} + d_{23}, d_{14} + d_{34})$$

Extreme Rays & Facets

$\mathcal{M}_n$ is a polyhedral cone.

- Facets $= 3\binom{3}{n}$ hyperplanes $d_{ik} + d_{kj} - d_{ij} = 0$.
- Avis (1980) shows that extreme rays are arbitrarily complex using graph metrics
- Let $G_{n,p}$ a random graph with n points and edge probability $P(ij \in G_{n,p} = \boldsymbol{p})$.
    - If for some $0 < \varepsilon < 1/5, n^{-1/5+\varepsilon} \leq \boldsymbol{p} \leq 1 - n^{-1/4+\varepsilon}$,
    - then the distance induced by G is an extreme ray of $\mathcal{M}_n$ with probability $1 - o(1)$.

Extreme Rays & Facets

$\mathcal{M}_n$ is a polyhedral cone.

- Facets $= 3\binom{3}{n}$ hyperplanes $d_{ik} + d_{kj} - d_{ij} = 0$.
- Avis (1980) shows that extreme rays are arbitrarily complex using graph metrics
- Let $G_{n,p}$ a random graph with n points and edge probability $P(ij \in G_{n,p} = \boldsymbol{p})$.
    - If for some $0 < \varepsilon < 1/5, n^{-1/5+\varepsilon} \leq \boldsymbol{p} \leq 1 - n^{-1/4+\varepsilon}$,
    - then the distance induced by G is an extreme ray of $\mathcal{M}_n$ with probability $1 - o(1)$.
- Grishukin (2005) characterizes the extreme rays of $\mathcal{M}_7$ (≥ 60.000)

Extreme Rays & Facets

$\mathcal{M}_n$ is a polyhedral cone.

- Facets $= 3\binom{3}{n}$ hyperplanes $d_{ik} + d_{kj} - d_{ij} = 0$.
- Avis (1980) shows that extreme rays are arbitrarily complex using graph metrics

$\mathcal{S}_n^+$ is a self-dual, homogeneous cone. **Overall far easier to study**:

- Facets are isomorphic to $\mathcal{S}_k^+$ for $k < n$
- Extreme rays exactly the p.s.d matrices of rank 1, $\mathbf{z}\mathbf{z}^T$.

Extreme Rays & Facets

$\mathcal{M}_n$ is a polyhedral cone.

- Facets $= 3\binom{3}{n}$ hyperplanes $d_{ik} + d_{kj} - d_{ij} = 0$.
- Avis (1980) shows that extreme rays are arbitrarily complex using graph metrics

$\mathcal{S}_n^+$ is a self-dual, homogeneous cone. **Overall far easier to study**:

- Facets are isomorphic to $\mathcal{S}_k^+$ for $k < n$
- Extreme rays exactly the p.s.d matrices of rank 1, $\mathbf{z}\mathbf{z}^T$.
  - $\rightarrow$ Eigendecomposition: if $K \in \mathcal{S}_n^+$ then $K = \sum_{i=1}^n \lambda_i \mathbf{z}_i \mathbf{z}_i^T$.
  - $\rightarrow$ Integral representations for p.d. kernels themselves (Bochner theorem)

Checking, Projection, Learning

Optimizing in $\boldsymbol{\mathcal{M}_n}$ is relatively difficult.

- Check if X is in $\boldsymbol{\mathcal{M}_n}$ requires up to $3\binom{3}{n}$ comparisons.
- Projection: triangle fixing algorithms (Brickell et al. (2008)), no convergence speed guarantee.
- No simple barrier function

Optimizing in $\boldsymbol{\mathcal{S}_n^+}$ is relatively easy.

- Check if X is in $\boldsymbol{\mathcal{S}_n^+}$ only requires finding minimal eigenvalue (`eigs`).
- Projection: threshold negative eigenvalues.
- $\log\det$ barrier, semidefinite programming

Checking, Projection, Learning

Optimizing in $\mathcal{M}_n$ is relatively difficult.

- Check if X is in $\mathcal{M}_n$ requires up to $3\binom{3}{n}$ comparisons.
- Projection: triangle fixing algorithms (Brickell et al. (2008)), no convergence speed guarantee.
- No simple barrier function

Optimizing in $\mathcal{S}_n^+$ is relatively easy.

- Check if X is in $\mathcal{S}_n^+$ only requires finding minimal eigenvalue (`eigs`).
- Projection: threshold negative eigenvalues.
- $\log\det$ barrier, semidefinite programming

"Real" metric learning in $\mathcal{M}_n$ is difficult, Mahalanobis learning in $\mathcal{S}_n^+$ is easier

Negative Definite Kernels

Convex cone of $n \times n$ **negative definite** kernels - dimension $\frac{n(n+1)}{2}$

$$\mathcal{N}_n = \{X \in \mathbb{R}^{n\times n} \,|\, X = X^T, \forall \mathbf{z} \in \mathbb{R}^n, \mathbf{z}^T\mathbf{1} = \mathbf{0}, \mathbf{z}^T X \mathbf{z} \leq 0\}$$

infinite linear inequalities; $\binom{2}{n}$ equalities

Negative Definite Kernels

Convex cone of $n \times n$ **negative definite** kernels - dimension $\frac{n(n+1)}{2}$

$$\mathcal{N}_n = \{X \in \mathbb{R}^{n \times n} \,|\, X = X^T, \forall \mathbf{z} \in \mathbb{R}^n, \mathbf{z^T 1 = 0}, \mathbf{z}^T X \mathbf{z} \leq 0\}$$

infinite linear inequalities; $\binom{2}{n}$ equalities

$$\psi \text{ n.d. kernel} \quad \Leftrightarrow \quad \forall\, n \in \mathbb{N}, \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}^n \quad \left[\psi(\mathbf{x}_i, \mathbf{x}_j)\right] \in \mathcal{N}_n$$

A few important results on Negative Definite Kernels

If ψ is a negative definite kernel on $\mathcal{X}$ then

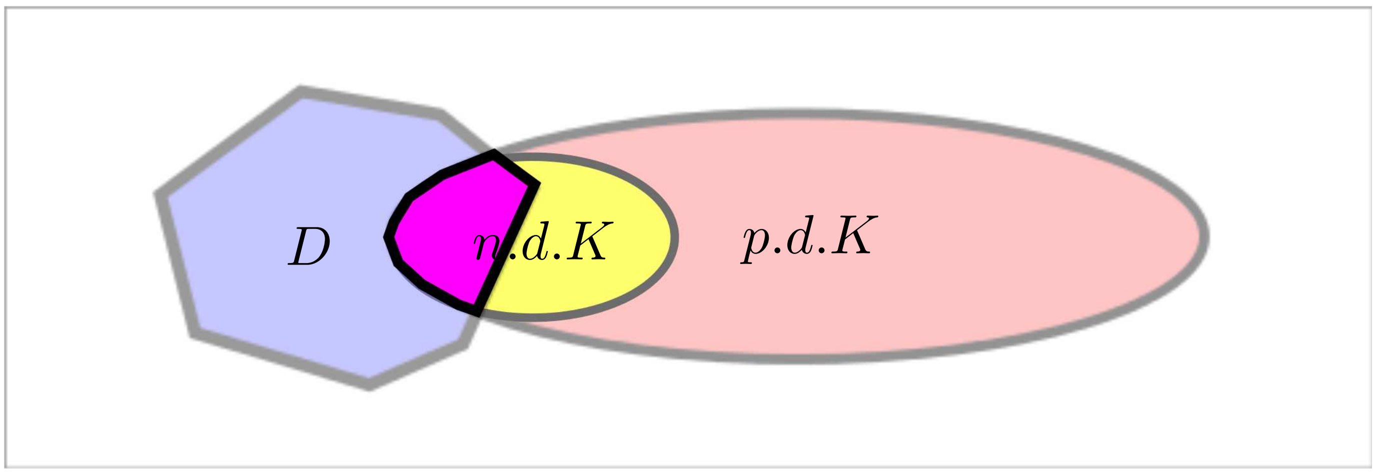
- $\exists$ a Hilbert space $\mathcal{H}$, a mapping $\mathbf{x} \mapsto \varphi_{\mathbf{x}} \in \mathcal{H}$, a real valued function f on $\mathcal{X}$ s.t.

$$\psi(\mathbf{x}, \mathbf{y}) = \|\varphi_x - \varphi_y\|^2 + f(x) + f(y)$$

- If $\forall \mathbf{x} \in \mathcal{X}, \psi(x, x) = 0$, then $f = 0$ and $\sqrt{\psi}$ is a semi-distance.
- If $\{\psi = 0\} = \{(\mathbf{x}, \mathbf{x}), \mathbf{x} \in \mathcal{X}\}$, then $\sqrt{\psi}$ is a distance.
- If $\psi(\mathbf{x}, \mathbf{x}) \geq 0$, then $1 < \alpha < 0$, ψ^{α} is negative definite.
- $\boldsymbol{k} \overset{\text{def}}{=} e^{-t\psi}$ is positive definite for all $t > 0$.

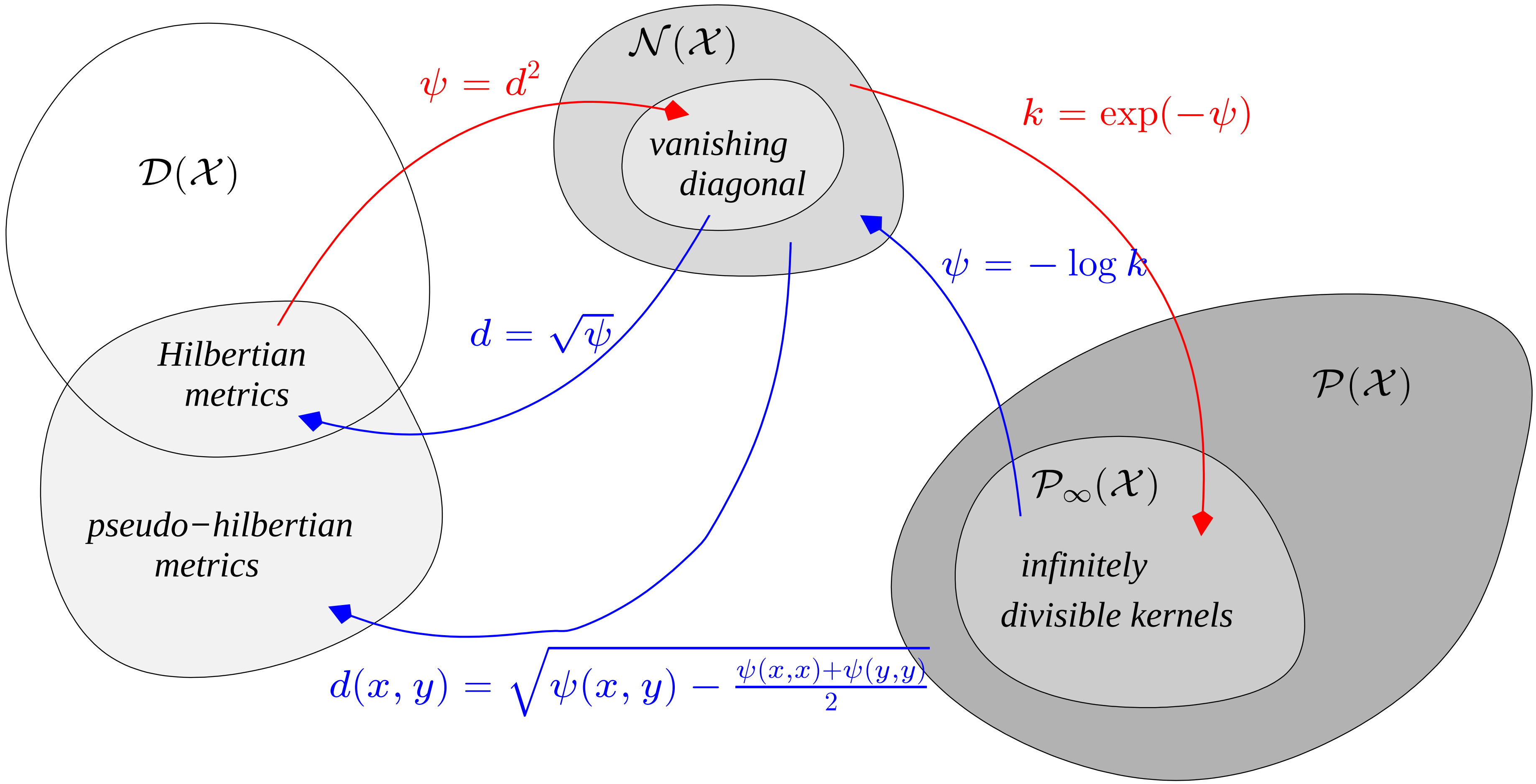
A Rough Sketch

We can now give a more precise meaning to



A Rough Sketch

using this diagram



Importance of this link

- One of the biggest practical issues with kernel methods is that of **diagonal dominance**.
  - Cauchy Schwartz: $\boldsymbol{k}(\mathbf{x}, \mathbf{y}) \leq \sqrt{\boldsymbol{k}(\mathbf{x}, \mathbf{x})\boldsymbol{k}(\mathbf{y}, \mathbf{y})}$

Importance of this link

- One of the biggest practical issues with kernel methods is that of **diagonal dominance**.
  - Cauchy Schwartz: $\boldsymbol{k}(\mathbf{x},\mathbf{y}) \leq \sqrt{\boldsymbol{k}(\mathbf{x},\mathbf{x})\boldsymbol{k}(\mathbf{y},\mathbf{y})}$
  - Diagonal dominance: $\boldsymbol{k}(\mathbf{x},\mathbf{y}) \ll \sqrt{\boldsymbol{k}(\mathbf{x},\mathbf{x})\boldsymbol{k}(\mathbf{y},\mathbf{y})}$

Importance of this link

- One of the biggest practical issues with kernel methods is that of **diagonal dominance**.
  - Cauchy Schwartz: $\boldsymbol{k}(\mathbf{x}, \mathbf{y}) \leq \sqrt{\boldsymbol{k}(\mathbf{x}, \mathbf{x})\boldsymbol{k}(\mathbf{y}, \mathbf{y})}$
  - Diagonal dominance: $\boldsymbol{k}(\mathbf{x}, \mathbf{y}) \ll \sqrt{\boldsymbol{k}(\mathbf{x}, \mathbf{x})\boldsymbol{k}(\mathbf{y}, \mathbf{y})}$
- If $\boldsymbol{k}$ is infinitely divisible, k^{α} with small α is
  - positive definite
  - less diagonally dominant

Importance of this link

- One of the biggest practical issues with kernel methods is that of **diagonal dominance**.
  - Cauchy Schwartz: $\boldsymbol{k}(\mathbf{x},\mathbf{y}) \leq \sqrt{\boldsymbol{k}(\mathbf{x},\mathbf{x})\boldsymbol{k}(\mathbf{y},\mathbf{y})}$
  - Diagonal dominance: $\boldsymbol{k}(\mathbf{x},\mathbf{y}) \ll \sqrt{\boldsymbol{k}(\mathbf{x},\mathbf{x})\boldsymbol{k}(\mathbf{y},\mathbf{y})}$
- If $\boldsymbol{k}$ is infinitely divisible, k^α with small α is
  - positive definite
  - less diagonally dominant
- This explain the **success** of
  - Gaussian kernels $e^{-t\|\mathbf{x}-\mathbf{y}\|^2}$
  - Laplace kernels $e^{-t\|\mathbf{x}-\mathbf{y}\|}$
- and arguably, the **failure** of many non-infinitely divisible kernels, because too difficult to tune.

Questions Worth Asking

Two questions:

Let d be a distance that is **not** negative definite.
is it possible that $e^{-t_1 d}$ is positive definite for some $t_1 \in \mathbb{R}$?

ε-infinite divisibility.
a distance d such that e^{-td} is positive definite for $t > \varepsilon$?

Questions Worth Asking

Two questions:

Let d be a distance that is **not** negative definite. is it possible that $e^{-t_1 d}$ is positive definite for some $t_1 \in \mathbb{R}$?

yes.

Examples exist. Stein distance (Sra, 2011) and Inverse generalized variance (C. et al., 2005) kernel for p.s.d matrices.

"ε-infinite divisibility". a distance d such that e^{-td} is positive definite for $t > \varepsilon$?

?

Positivity & Combinatorial Distances

Structured Objects

- Objects in a countable set
  - variable length strings, trees, graphs, permutations
- Constrained vectors
  - Positive vectors, histograms
- Vectors of different sizes
  - variable length time series

Structured Objects

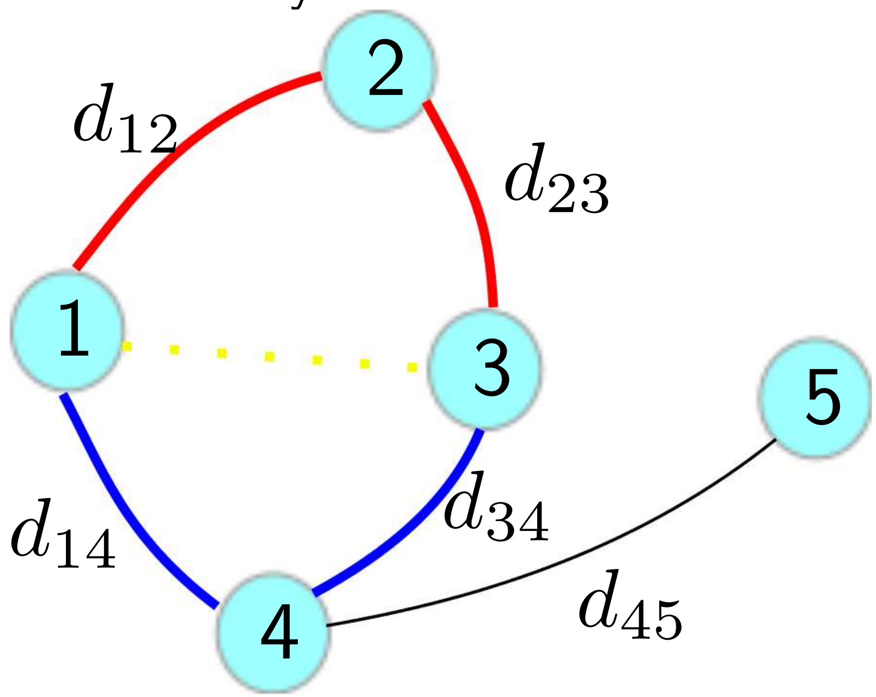
- Objects in a countable set
  - variable length strings, trees, graphs, sets
- Constrained vectors
  - Positive vectors, histograms
- Vectors of different sizes
  - variable length time series

How can we define a **kernel** or a **distance** on such sets?

in most cases, applying standard distances on $\mathbb{R}^n$ or even $\mathbb{N}^n$ is meaningless

Back to fundamentals

- **Distances** are **optimal** by nature, and quantify **shortest length paths**.
  - Graph-metrics are defined that way

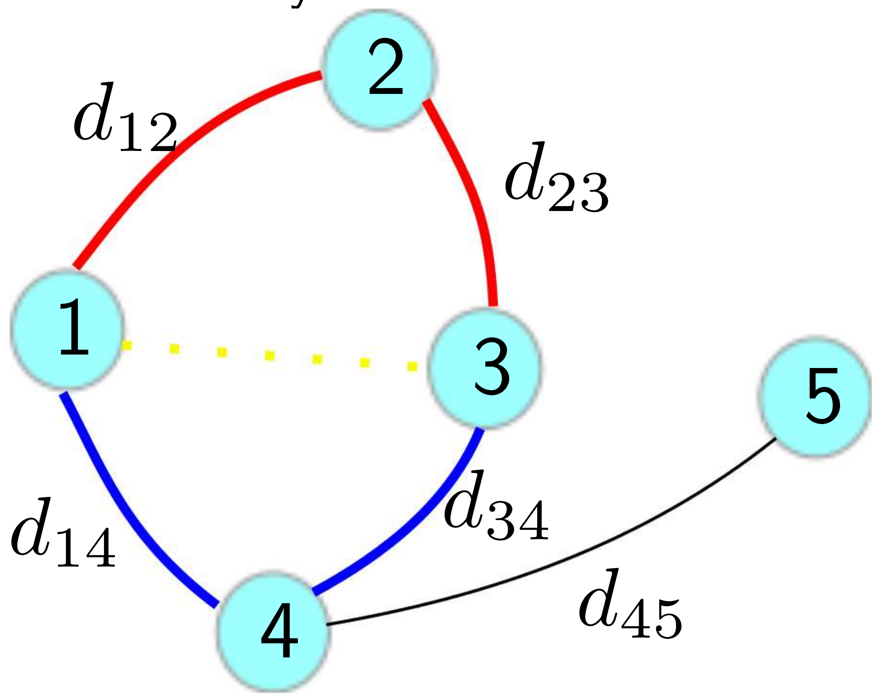


  - **Triangle inequalities** are defined precisely to enforce this optimality

$$d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$$

Back to fundamentals

- **Distances** are **optimal** by nature, and quantify **shortest length paths**.
  - Graph-metrics are defined that way



  - **Triangle inequalities** are defined precisely to enforce this optimality

$$d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$$

$\rightarrow$ many **distances** on structured objects rely on **optimization**

Back to fundamentals

- **p.d. kernels** are additive by nature
  - $\boldsymbol{k}$ is positive definite $\Leftrightarrow \exists\varphi : \mathcal{X} \rightarrow \mathcal{H}$ such that

$$\boldsymbol{k}(\mathbf{x},\mathbf{y}) = \langle\varphi(x),\varphi(y)\rangle_{\mathcal{H}}.$$

- $X \in \boldsymbol{\mathcal{S}_n^+} \Leftrightarrow \exists L \in \mathbb{R}^{n\times n} | X = L^T L.$

Back to fundamentals

- **p.d. kernels** are additive by nature
  - $\boldsymbol{k}$ is positive definite $\Leftrightarrow \exists\varphi : \mathcal{X} \rightarrow \mathcal{H}$ such that

$$\boldsymbol{k}(\mathbf{x}, \mathbf{y}) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}.$$

- $X \in \boldsymbol{\mathcal{S}_n^+} \Leftrightarrow \exists L \in \mathbb{R}^{n\times n} | X = L^T L.$

→ many **kernels** on structured objects
rely on defining **explicitly** (possibly infinite) feature vectors

very large literature on this subject which we will not address here.

Combinatorial Distances

- To define a **distance**, an approach which has been repeatedly used is to,
  - Consider two inputs $\mathbf{x}, \mathbf{y}$,
  - Define a **countable** set of **mappings** from $\mathbf{x}$ to $\mathbf{y}$, $T(\mathbf{x}, \mathbf{y})$
  - Define a **cost** $c(\tau)$ for each element τ of $T(\mathbf{x}, \mathbf{y})$.
  - Define a distance between $\mathbf{x}, \mathbf{y}$ as

$$d(\mathbf{x}, \mathbf{y}) = \min_{\tau \in T(\mathbf{x}, \mathbf{y})} c(\tau)$$

Combinatorial Distances

- To define a **distance**, an approach which has been repeatedly used is to,
  - Consider two inputs $\mathbf{x}, \mathbf{y}$,
  - Define a **countable** set of **mappings** from $\mathbf{x}$ to $\mathbf{y}$, $T(\mathbf{x}, \mathbf{y})$
  - Define a **cost** $c(\tau)$ for each element τ of $T(\mathbf{x}, \mathbf{y})$.
  - Define a distance between $\mathbf{x}, \mathbf{y}$ as

$$d(\mathbf{x}, \mathbf{y}) = \min_{\tau \in T(\mathbf{x}, \mathbf{y})} c(\tau)$$

- **Symmetry, definiteness and triangle inequalities** depend on c and T.

Combinatorial Distances

- To define a **distance**, an approach which has been repeatedly used is to,
  - Consider two inputs $\mathbf{x}, \mathbf{y}$,
  - Define a **countable** set of **mappings** from $\mathbf{x}$ to $\mathbf{y}$, $T(\mathbf{x}, \mathbf{y})$
  - Define a **cost** $c(\tau)$ for each element τ of $T(\mathbf{x}, \mathbf{y})$.
  - Define a distance between $\mathbf{x}, \mathbf{y}$ as

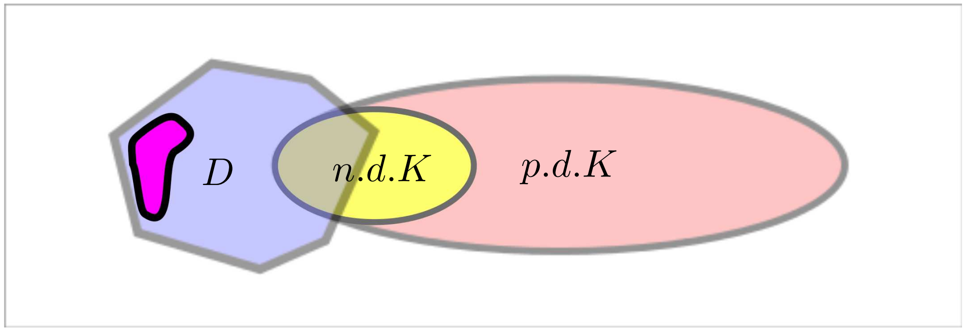
$$d(\mathbf{x}, \mathbf{y}) = \min_{\tau \in T(\mathbf{x}, \mathbf{y})} c(\tau)$$

- **Symmetry, definiteness and triangle inequalities** depend on c and T.
- In many cases, T is endowed with a dot product, $c(\tau) = \langle \tau, \theta \rangle$ for some θ.

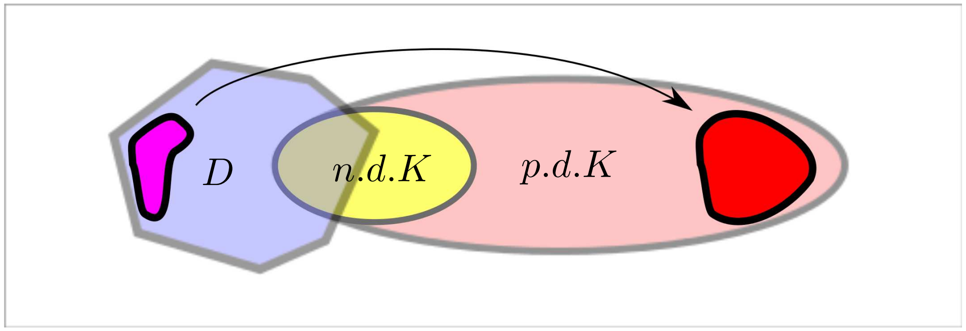
Combinatorial Distances are not Negative Definite

$$d(\mathbf{x}, \mathbf{y}) = \min_{\tau \in T(\mathbf{x},\mathbf{y})} c(\tau)$$

- In most cases such distances are **not** negative definite



- Can we use them to define kernels?

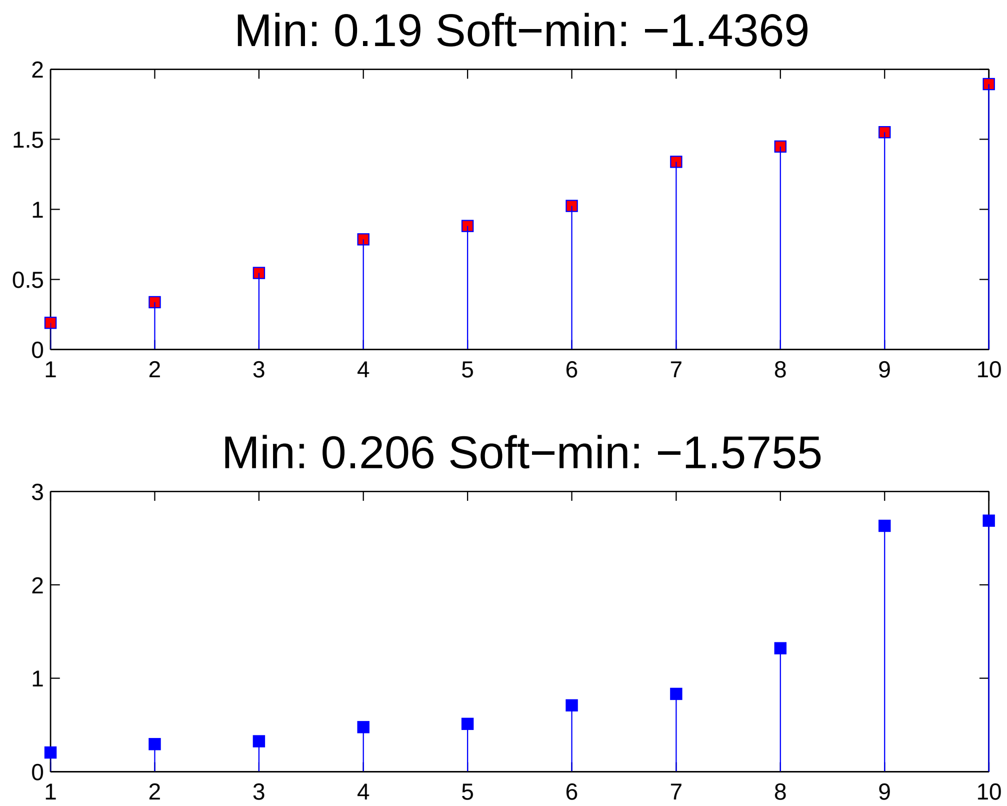


- **Yes** so far, using always the **same technique**.

An alternative definition of minimality

for a family of numbers $a_n, n \in \mathbb{N}$,

$$\text{soft-min} a_n = -\log \sum_n e^{-a_n}$$



Soft-min of costs - Generating Functions

$$d(\mathbf{x},\mathbf{y}) = \min_{\tau \in T(\mathbf{x},\mathbf{y})} c(\tau)$$

e^{-d} is **not** positive definite in the general case

Soft-min of costs - Generating Functions

$$d(\mathbf{x}, \mathbf{y}) = \min_{\tau \in T(\mathbf{x},\mathbf{y})} c(\tau)$$

e^{-d} is **not** positive definite in the general case

$$\delta(\mathbf{x}, \mathbf{y}) = \underset{\tau \in T(\mathbf{x},\mathbf{y})}{\text{soft-min}} \quad c(\tau)$$

$e^{-\delta}$ has been proved to be **positive definite** in all known cases

Soft-min of costs - Generating Functions

$$d(\mathbf{x},\mathbf{y}) = \min_{\tau \in T(\mathbf{x},\mathbf{y})} c(\tau)$$

e^{-d} is **not** positive definite in the general case

$$\delta(\mathbf{x},\mathbf{y}) = \underset{\tau \in T(\mathbf{x},\mathbf{y})}{\text{soft-min}} \quad c(\tau)$$

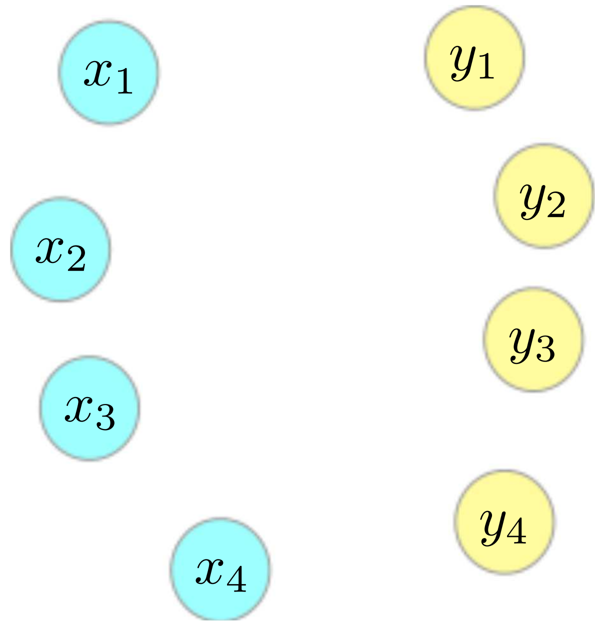
$e^{-\delta}$ has been proved to be **positive definite** in all known cases

$$e^{-\delta(\mathbf{x},\mathbf{y})} = \sum_{\tau \in T(\mathbf{x},\mathbf{y})} e^{-\langle \tau, \theta \rangle} = G_{T(\mathbf{x},\mathbf{y})}(\theta)$$

$G_{T(\mathbf{x},\mathbf{y})}$ is the **generating function** of the set of all mappings between **x** and **y**.

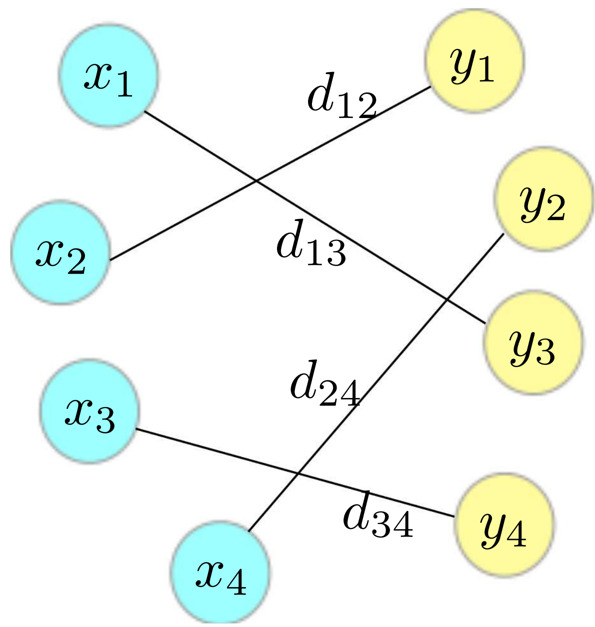
Example: Optimal assignment distance between two sets

- **Input**: $\mathbf{x} = \{x_1, \cdots, x_n\}, \mathbf{y} = \{y_1, \cdots, y_n\} \in \mathcal{X}^n$



Example: Optimal assignment distance between two sets

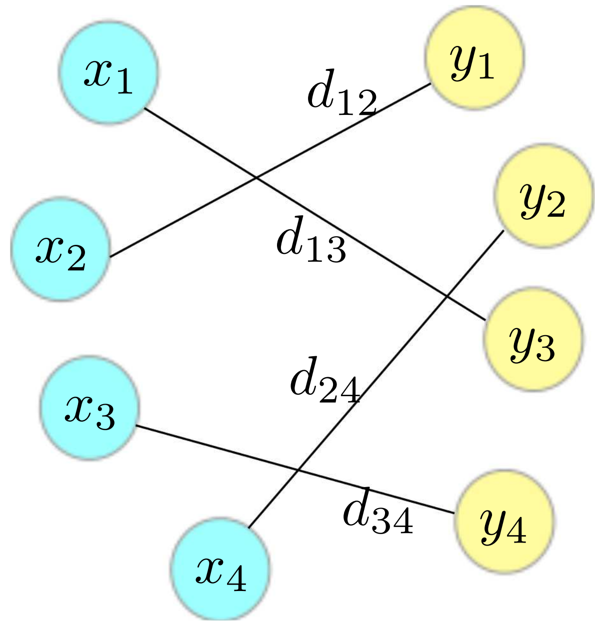
- **Input**: $\mathbf{x} = \{x_1, \cdots, x_n\}, \mathbf{y} = \{y_1, \cdots, y_n\} \in \mathcal{X}^n$



- **cost parameter**: distance d on $\mathcal{X}$. **mapping variable**: permutation σ in S_n
- **cost**: $\sum_{i=1}^{n} d(x_i, y_{\sigma(i)})$.

Example: Optimal assignment distance between two sets

- **Input**: $\mathbf{x} = \{x_1, \cdots, x_n\}, \mathbf{y} = \{y_1, \cdots, y_n\} \in \mathcal{X}^n$



- **cost parameter**: distance d on $\mathcal{X}$. **mapping variable**: permutation σ in S_n.
- **cost**: $\sum_{i=1}^{n} d(x_i, y_{\sigma(i)}) = \langle P_\sigma, D\rangle$ where $D = [d(x_i, y_j)]$

$$\boxed{d_{\text{Assig.}}(\mathbf{x}, \mathbf{y}) = \min_{\sigma \in S_n} \sum_{i=1}^{n} d(x_i, y_{\sigma(i)}) = \min_{\sigma \in S_n} \langle P_\sigma, D\rangle}$$

Example: Optimal assignment distance between two sets

$$\boxed{d_{\text{Assig.}}(\mathbf{x},\mathbf{y}) = \min_{\sigma\in S_n}\sum_{i=1}^{n} d(x_i, y_{\sigma(i)}) = \min_{\sigma\in S_n}\langle P_\sigma, D\rangle}$$

define $k = e^{-d}$. If k is positive definite on $\mathcal{X}$ then

$$\boxed{k_{\text{Perm}}(\mathbf{x},\mathbf{y}) = \sum_{\sigma\in S_n} e^{-\langle P_\sigma, D\rangle} = \text{Permanent}[k(x_i, y_j)]}$$

is positive definite (C. 2007). $e^{-d_{\text{Assig.}}}$ is not (Frohlich et al. 2005, Vert 2008).

Example: Optimal alignment between two strings

- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathcal{X}^n$, $\mathcal{X}$ finite

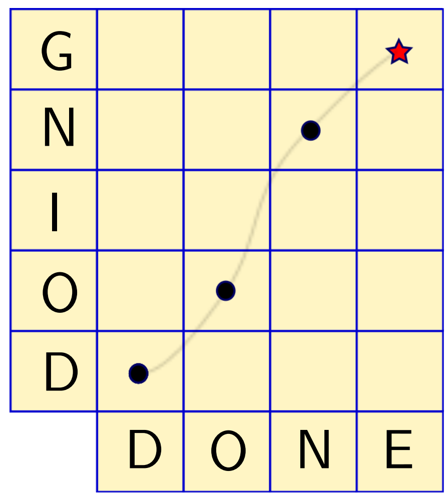
$$\mathbf{x} = \texttt{DOING},\ \mathbf{y} = \texttt{DONE}$$

Example: Optimal alignment between two strings

- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathcal{X}^n$, $\mathcal{X}$ finite

$$\mathbf{x} = \mathtt{DOING},\ \mathbf{y} = \mathtt{DONE}$$

- **mapping variable**: alignment $\pi = \begin{pmatrix} \pi_1(1) & \cdots & \pi_1(q) \\ \pi_2(1) & \cdots & \pi_2(q) \end{pmatrix}$. (increasing path)

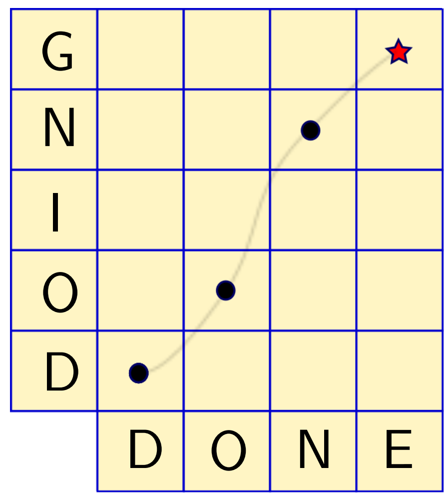


Example: Optimal alignment between two strings

- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathcal{X}^n$, $\mathcal{X}$ finite

$$\mathbf{x} = \texttt{DOING},\ \mathbf{y} = \texttt{DONE}$$

- **mapping variable**: alignment $\pi = \begin{pmatrix} \pi_1(1) & \cdots & \pi_1(q) \\ \pi_2(1) & \cdots & \pi_2(q) \end{pmatrix}$. (increasing path)



- **cost parameter**: distance d on $\mathcal{X}$ + gap function $g : \mathbb{N} \to \mathbb{R}$.

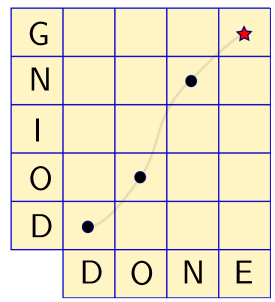
- $c(\pi) = \sum_{i=1}^{|\pi|} d(x_{\pi_1(i)}, y_{\pi_2(i)}) + \sum_{i=1}^{|\pi|-1} g(\pi_1(i+1) - \pi_1(i)) + g(\pi_2(i+1) - \pi_2(i))$

Example: Optimal alignment between two strings

- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathcal{X}^n$, $\mathcal{X}$ finite

$$\mathbf{x} = \texttt{DOING}, \ \mathbf{y} = \texttt{DONE}$$

- **mapping variable**: alignment $\pi = \begin{pmatrix} \pi_1(1) & \cdots & \pi_1(q) \\ \pi_2(1) & \cdots & \pi_2(q) \end{pmatrix}$. (increasing path)



- **cost parameter**: distance d on $\mathcal{X}$ + gap function $g : \mathbb{N} \to \mathbb{R}$.
- $c(\pi) = \sum_{i=1}^{|\pi|} d(x_{\pi_1(i)}, y_{\pi_2(i)}) + \sum_{i=1}^{|\pi|-1} g(\pi_1(i+1) - \pi_1(i)) + g(\pi_2(i+1) - \pi_2(i))$

$$\boxed{d_{\text{align}}(\mathbf{x}, \mathbf{y}) = \min_{\pi \in \text{Alignments}} c(\pi)}$$

Example: Optimal alignment between two strings

$$\boxed{d_{\text{align}}(\mathbf{x}, \mathbf{y}) = \min_{\pi \in \text{Alignments}} c(\pi)}$$

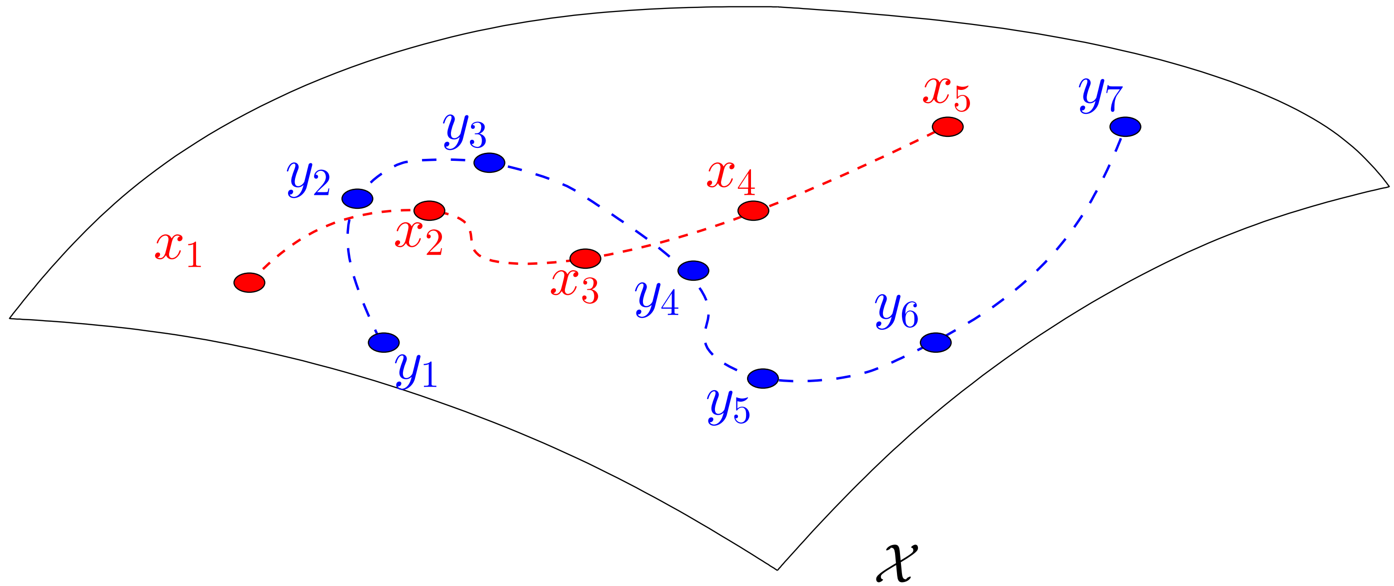
define $k = e^{-d}$. If k is positive definite on $\mathcal{X}$ then

$$\boxed{k_{\text{LA}}(\mathbf{x}, \mathbf{y}) = \sum_{\pi \in \text{Alignments}} e^{-c(\pi)}}$$

is positive definite (Saigo et al. 2003).

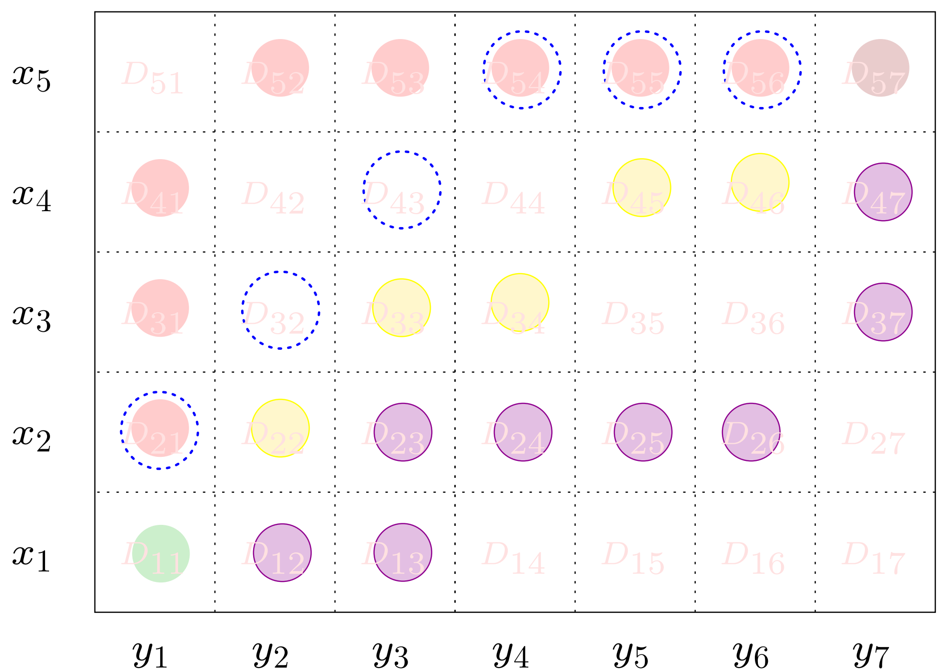
Example: Optimal time warping between two time series

- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathbb{R}^n$



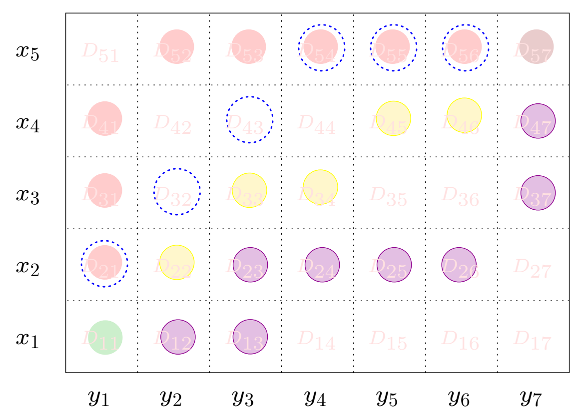
Example: Optimal time warping between two time series

- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathbb{R}^n$
- **mapping variable**: $\pi = \begin{pmatrix} \pi_1(1) & \cdots & \pi_1(q) \\ \pi_2(1) & \cdots & \pi_2(q) \end{pmatrix}$. (increasing **contiguous** path)



Example: Optimal time warping between two time series

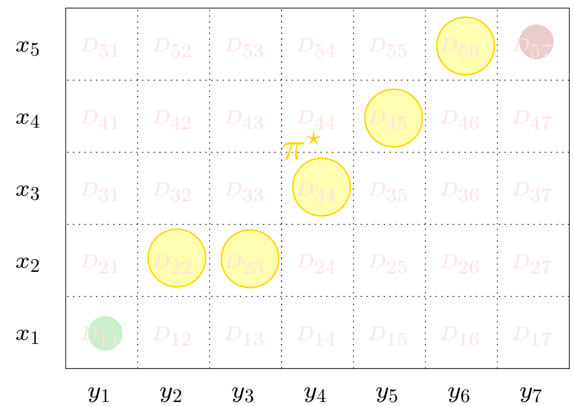
- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathbb{R}^n$
- **mapping variable**: $\pi = \begin{pmatrix} \pi_1(1) & \cdots & \pi_1(q) \\ \pi_2(1) & \cdots & \pi_2(q) \end{pmatrix}$. (increasing **contiguous** path)



- **cost parameter**: distance d on $\mathcal{X}$. **cost**: $c(\pi) = \sum_{i=1}^{|\pi|} d(x_{\pi_1(i)}, y_{\pi_2(i)})$

Example: Optimal time warping between two time series

- **Input**: $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_m) \in \mathbb{R}^n$
- **mapping variable**: $\pi = \begin{pmatrix} \pi_1(1) & \cdots & \pi_1(q) \\ \pi_2(1) & \cdots & \pi_2(q) \end{pmatrix}$. (increasing **contiguous** path)



- **cost parameter**: distance d on $\mathcal{X}$. **cost**: $c(\pi) = \sum_{i=1}^{|\pi|} d(x_{\pi_1(i)}, y_{\pi_2(i)})$

$$\boxed{d_{\mathsf{DTW}}(\mathbf{x}, \mathbf{y}) = \min_{\pi \in \mathsf{Alignments}} c(\pi)}$$

Example: Optimal alignment between two strings

$$\boxed{d_{\mathsf{DTW}}(\mathbf{x},\mathbf{y}) = \min_{\pi\in\mathsf{Alignments}} c(\pi)}$$

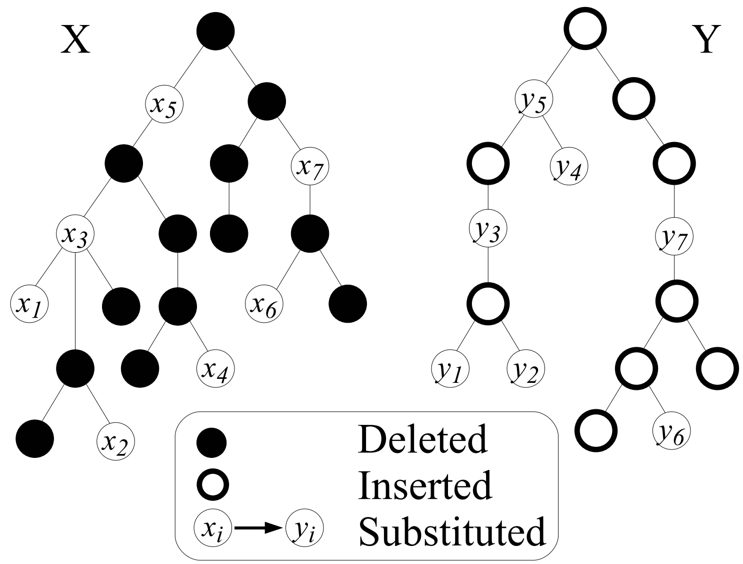
define $k = e^{-d}$. If k is positive definite and geometrically divisible on $\mathcal{X}$ then

$$\boxed{k_{\mathsf{GA}}(\mathbf{x},\mathbf{y}) = \sum_{\pi\in\mathsf{Alignments}} e^{-c(\pi)}}$$

is positive definite (C. et al. 2007, C. 2011)

Example: Edit-distance between two trees

- **Input**: two labeled trees $\mathbf{x}, \mathbf{y}$.
- **mapping variable**: sequence of substitutions/deletions/insertions of vertices

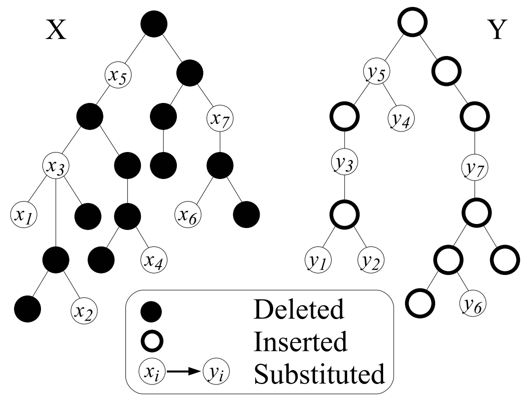


- **cost parameter**: γ distance between labels and cost for deletion/insertion

$$\boxed{d_{\mathsf{TreeEdit}}(\mathbf{x},\mathbf{y}) = \min_{\sigma \in \mathsf{EditScripts}(\mathbf{x},\mathbf{y})} \sum \gamma(\sigma_i)}$$

Example: Edit-distance between two trees

- **Input**: two labeled trees $\mathbf{x}, \mathbf{y}$.
- **mapping variable**: sequence of substitutions/deletions/insertions of vertices



- **cost parameter**: γ distance between labels and cost for deletion/insertion

$$\boxed{d_{\mathsf{TreeEdit}}(\mathbf{x}, \mathbf{y}) = \min_{\sigma \in \mathsf{EditScripts}(\mathbf{x},\mathbf{y})} \sum \gamma(\sigma_i)}$$

- Positive definiteness of the generating function (if $e^{-\gamma}$) p.d. proved by Shin & Kuboyama 2008; Shin, C., Kuboyama 2011.

Example: Transportation distance between discrete histograms

- **Input**: two integer histograms $\mathbf{x}, \mathbf{y} \in \mathbb{N}^d$ such that $\sum_{i=1}^d x_i = \sum_{i=1}^d y_i = N$

- **mapping**: transportation matrices $U(r,c) = \{X \in \mathbb{N}^{d\times d} | X\mathbf{1}_d = \mathbf{x}, X^T\mathbf{1}_d = \mathbf{y}\}$

- **cost parameter**: M distance matrix in $\mathcal{M}_d$.

$$\boxed{d_W(\mathbf{x}, \mathbf{y}) = \min_{X \in U(r,c)} \langle X, M \rangle}$$

Example: Transportation distance between discrete histograms

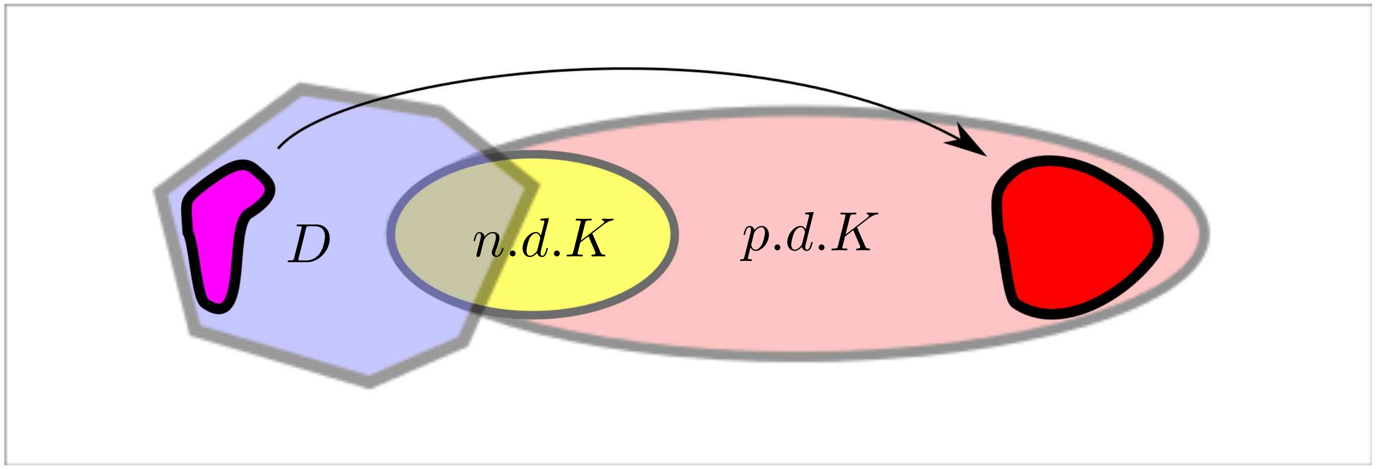
$$\boxed{d_W(\mathbf{x}, \mathbf{y}) = \min_{X \in U(r,c)} \langle X, M \rangle}$$

define $k_{ij} = e^{-m_{ij}}$. If $[k_{ij}]$ is positive definite on $\mathcal{X}$ then

$$\boxed{k_M(\mathbf{x}, \mathbf{y}) = \sum_{X \in U(r,c)} e^{-\langle X, M \rangle}}$$

is positive definite (C., submitted).

To wrap up



$$d(\mathbf{x},\mathbf{y}) = \min_{\tau\in T(\mathbf{x},\mathbf{y})} c(\tau), \quad \delta(\mathbf{x},\mathbf{y}) = \underset{\tau\in T(\mathbf{x},\mathbf{y})}{\text{soft-min}} \; c(\tau)$$

$$\boxed{e^{-\delta(\mathbf{x},\mathbf{y})} = \sum_{\tau\in T(\mathbf{x},\mathbf{y})} e^{-\langle\tau,\theta\rangle} = G_{T(\mathbf{x},\mathbf{y})}(\theta) \text{ is positive definite in many (all) cases.}}$$

Open problems

- $\exists$ unified framework?
  - **Convolution kernels** (Haussler, 1998)
  - **Mapping kernels** (Shin & Kuboyama 2008) were an important addition
  - Extension to **Countable mapping kernels** (Shin 2011)
  - Extension to symmetric functions (not just $e^{\cdot}$) (Shin 2011).
- To speed up computations, possible to restrict the sum to subset of $T(\mathbf{x}, \mathbf{y})$?
  - C. 2011 with DTW.
  - C. submitted with transportation distances