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.
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
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)
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 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.
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)...
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
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)
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)
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 $\mathcal{X}$,

$$
\begin{align*}
    d & : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \\
    (x, y) & \mapsto d(x, y)
\end{align*}
$$

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

- $d(x, y) = d(y, x)$, symmetry
- $d(x, y) = 0 \iff x = y$, definiteness
- $d(x, z) \leq d(x, y) + d(y, z)$, triangle inequality
Kernels (Symmetric & Positive Definite)

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

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$$

$$(x, y) \mapsto k(x, y)$$

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

- $k(x, y) = k(y, x)$, symmetry

and $\forall n \in \mathbb{N}, \{x_1, \ldots, x_n\} \in \mathcal{X}^n, c \in \mathbb{R}^n$

- $\sum_{i=1}^n c_i c_j k(x_i, x_j) \geq 0$
Matrices

Convex cone of $n \times n$ distance matrices - dimension $\frac{n(n-1)}{2}$

$\mathcal{M}_n = \{ X \in \mathbb{R}^{n \times n} \mid x_{ii} = 0; \text{ for } i < j, x_{ij} > 0; x_{ik} + x_{kj} - x_{ij} \geq 0 \}$

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

Convex cone of $n \times n$ **distance** matrices - dimension $\frac{n(n-1)}{2}$

$$\mathcal{M}_n = \{X \in \mathbb{R}^{n \times n} | x_{ii} = 0; \text{ for } i \neq j, x_{ij} > 0; x_{ik} + x_{kj} - x_{ij} \geq 0\}$$

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

Convex cone of $n \times n$ **p.s.d. matrices** - dimension $\frac{n(n+1)}{2}$

$$\mathcal{S}_n^+ = \{X \in \mathbb{R}^{n \times n} | X = X^T; \forall z \in \mathbb{R}^n, z^TXz \geq 0\}$$

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

\[
\begin{bmatrix}
0 & x & y \\
x & 0 & z \\
y & z & 0
\end{bmatrix}
\]

\[\partial S^2\]

\[\sqrt{2}\beta\]

image: Dattoro
$$d \text{ distance} \iff \forall n \in \mathbb{N}, \{x_1, \ldots, x_n\} \in \mathcal{X}^n \quad [d(x_i, x_j)] \in \mathcal{M}_n$$

$$k \text{ kernel} \iff \forall n \in \mathbb{N}, \{x_1, \ldots, x_n\} \in \mathcal{X}^n \quad [k(x_i, x_j)] \in \mathcal{S}^+_n$$
\( M_n \) 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.
\( M_n \) is a polyhedral cone.

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

\[
d_{13} = \min(d_{12} + d_{23}, d_{14} + d_{34})
\]
$\mathcal{M}_n$ 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 $P(ij \in G_{n,p} = p)$.
  - If for some $0 < \varepsilon < 1/5$, $n^{-1/5+\varepsilon} \leq p \leq 1 - n^{-1/4+\varepsilon}$,
  - then the distance induced by $G$ is an extreme ray of $\mathcal{M}_n$ with probability $1 - o(1)$. 
$\mathcal{M}_n$ 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 $P(\ ij \in G_{n,p} = p)$.
  - If for some $0 < \epsilon < 1/5$, $n^{-1/5+\epsilon} \leq p \leq 1 - n^{-1/4+\epsilon}$,
  - 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 $\mathcal{M}_7$ ($\geq 60,000$)
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, $zz^T$. 
**Extreme Rays & Facets**

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

- Facets = $3 \binom{n}{3}$ 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, $zz^T$.
  - $\rightarrow$ Eigendecomposition: if $K \in \mathcal{S}_n^+$ then $K = \sum_{i=1}^{n} \lambda_i z_i z_i^T$.
  - $\rightarrow$ Integral representations for p.d. kernels themselves (Bochner theorem)
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 ($\text{eigs}$).

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

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

infinite linear inequalities; \binom{2}{n} equalities
Convex cone of $n \times n$ negative definite kernels - dimension $\frac{n(n+1)}{2}$

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

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

$\psi$ n.d. kernel $\iff \forall n \in \mathbb{N}, \{x_1, \cdots, x_n\} \in \mathcal{X}^n \quad [\psi(x_i, x_j)] \in \mathcal{N}_n$
A few important results on Negative Definite Kernels

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

- There exists a Hilbert space $\mathcal{H}$, a mapping $x \mapsto \varphi_x \in \mathcal{H}$, a real valued function $f$ on $\mathcal{X}$ such that
  \[ \psi(x, y) = \|\varphi_x - \varphi_y\|^2 + f(x) + f(y) \]

- If $\forall x \in \mathcal{X}, \psi(x, x) = 0$, then $f = 0$ and $\sqrt{\psi}$ is a semi-distance.

- If $\{\psi = 0\} = \{(x, x), x \in \mathcal{X}\}$, then $\sqrt{\psi}$ is a distance.

- If $\psi(x, x) \geq 0$, then $1 < \alpha < 0$, $\psi^\alpha$ is negative definite.

- $k \overset{\text{def}}{=} e^{-t\psi}$ is positive definite for all $t > 0$. 
We can now give a more precise meaning to
A Rough Sketch

using this diagram

\[ d(x, y) = \sqrt{\psi(x, y) - \frac{\psi(x, x) + \psi(y, y)}{2}} \]

\[ k = \exp(-\psi) \]

\[ \psi = -\log k \]

\[ \psi = d^2 \]
Importance of this link

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

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

- Cauchy Schwartz: \( k(x, y) \leq \sqrt{k(x, x)k(y, y)} \)
- Diagonal dominance: \( k(x, y) \ll \sqrt{k(x, x)k(y, y)} \)

If \( k \) is infinitely divisible, \( k^\alpha \) with small \( \alpha \) is

- positive definite
- less diagonally dominant
One of the biggest practical issues with kernel methods is that of \textbf{diagonal dominance}.

- Cauchy Schwartz: $k(x, y) \leq \sqrt{k(x, x)k(y, y)}$
- Diagonal dominance: $k(x, y) \ll \sqrt{k(x, x)k(y, y)}$

- If $k$ is infinitely divisible, $k^{\alpha}$ with small $\alpha$ is
  - positive definite
  - less diagonally dominant

This explain the \textbf{success} of

- Gaussian kernels $e^{-t\|x-y\|^2}$
- Laplace kernels $e^{-t\|x-y\|}$

and arguably, the \textbf{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 $t_1 \in \mathbb{R}$?

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

Let $d$ be a distance that is not negative definite. Is it possible that $e^{-t_1d}$ 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.

“$\varepsilon$-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
• **Distances** are *optimal* by nature, and quantify *shortest length paths*.
  
  ◦ Graph-metrics are defined that way

  ![Graph with distances](image)

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

\[
d(x, y) \leq d(x, z) + d(z, y)
\]
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

\[ d(x, y) \leq d(x, z) + d(z, y) \]

→ many **distances** on structured objects rely on **optimization**
• **p.d. kernels** are additive by nature
  - $k$ is positive definite $\iff \exists \varphi : \mathcal{X} \to \mathcal{H}$ such that
    $$k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}.$$  

• $X \in \mathcal{S}_n^+ \iff \exists L \in \mathbb{R}^{n \times n} | X = L^T L.$
• **p.d. kernels** are additive by nature
  
  ○ $k$ is positive definite $\iff \exists \varphi : \mathcal{X} \rightarrow \mathcal{H}$ such that

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

  → 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.

• $X \in \mathcal{S}_n^+ \iff \exists L \in \mathbb{R}^{n \times n} | X = L^T L.$
To define a distance, an approach which has been repeatedly used is to,

- Consider two inputs $x, y$,
- Define a countable set of mappings from $x$ to $y$, $T(x, y)$
- Define a cost $c(\tau)$ for each element $\tau$ of $T(x, y)$.
- Define a distance between $x, y$ as

$$d(x, y) = \min_{\tau \in T(x, y)} c(\tau)$$
To define a **distance**, an approach which has been repeatedly used is to,

- Consider two inputs $x, y$,
- Define a **countable** set of mappings from $x$ to $y$, $T(x, y)$
- Define a **cost** $c(\tau)$ for each element $\tau$ of $T(x, y)$.
- Define a distance between $x, y$ as

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

**Symmetry, definiteness and triangle inequalities** depend on $c$ and $T$. 
To define a **distance**, an approach which has been repeatedly used is to,

- Consider two inputs \( x, y \),
- Define a **countable** set of **mappings** from \( x \) to \( y \), \( T(x, y) \)
- Define a **cost** \( c(\tau) \) for each element \( \tau \) of \( T(x, y) \).
- Define a distance between \( x, y \) as

\[
    d(x, y) = \min_{\tau \in T(x, y)} c(\tau)
\]

**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 \( \theta \).
Combinatorial Distances are not Negative Definite

\[ d(x, y) = \min_{\tau \in T(x, y)} c(\tau) \]

- In most cases such distances are \textbf{not} negative definite

- Can we use them to define kernels?

- \textbf{Yes} so far, using always the \textit{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

\[ d(x, y) = \min_{\tau \in T(x, y)} c(\tau) \]

\[ e^{-d} \text{ is not positive definite in the general case} \]
Soft-min of costs - Generating Functions

\[ d(x, y) = \min_{\tau \in T(x, y)} c(\tau) \]

\[ e^{-d} \text{ is not positive definite in the general case} \]

\[ \delta(x, y) = \text{soft-min}_{\tau \in T(x, y)} c(\tau) \]

\[ e^{-\delta} \text{ has been proved to be positive definite in all known cases} \]
Soft-min of costs - Generating Functions

\[ d(x, y) = \min_{\tau \in T(x, y)} c(\tau) \]

\( e^{-d} \) is not positive definite in the general case

\[ \delta(x, y) = \text{soft-min}_{\tau \in T(x, y)} c(\tau) \]

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

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

\( G_{T(x, y)} \) is the generating function of the set of all mappings between \( x \) and \( y \).
Example: Optimal assignment distance between two sets

- **Input:** $x = \{x_1, \cdots, x_n\}, y = \{y_1, \cdots, y_n\} \in \mathcal{X}^n$
Example: Optimal assignment distance between two sets

- **Input**: \( x = \{x_1, \ldots, x_n\}, y = \{y_1, \ldots, y_n\} \in \mathcal{X}^n \)

  ![Diagram of assignment distance](image)

- **cost parameter**: distance \( d \) on \( \mathcal{X} \). **mapping variable**: permutation \( \sigma \) in \( S_n \)

- **cost**: \( \sum_{i=1}^{n} d(x_i, y_{\sigma(i)}) \).
Example: Optimal assignment distance between two sets

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

- **Cost parameter:** distance \( d \) on \( \mathcal{X} \). **Mapping variable:** permutation \( \sigma \) in \( S_n \).

- **Cost:** \( \sum_{i=1}^n d(x_i, y_{\sigma(i)}) = \langle P_\sigma, D \rangle \) where \( D = [d(x_i, y_j)] \)

\[
d_{\text{Assig.}}(x, y) = \min_{\sigma \in S_n} \sum_{i=1}^n d(x_i, y_{\sigma(i)}) = \min_{\sigma \in S_n} \langle P_\sigma, D \rangle
\]
Example: Optimal assignment distance between two sets

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

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

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

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

\[
x = \text{DOING}, \quad y = \text{DONE}
\]
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

  \[ x = \text{DOING}, \ y = \text{DONE} \]

- **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
  
  \[ x = \text{DOING}, \ y = \text{DONE} \]

- **mapping variable:** alignment \( \pi = \left( \pi_1(1) \cdots \pi_1(q) \right) \left( \pi_2(1) \cdots \pi_2(q) \right) \). (increasing path)

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

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

  \[ x = \text{DOING}, y = \text{DONE} \]

- **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 \( g : \mathbb{N} \to \mathbb{R} \).

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

\[ d_{\text{align}}(x, y) = \min_{\pi \in \text{Alignments}} c(\pi) \]
Example: Optimal alignment between two strings

\[ d_{\text{align}}(x, y) = \min_{\pi \in \text{Alignments}} c(\pi) \]

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

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

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

- **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:** $c(\pi) = \sum_{i=1}^{\lvert \pi \rvert} d(x_{\pi_1(i)}, y_{\pi_2(i)})$
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 \( X \). **Cost**: 
  \[
  c(\pi) = \sum_{i=1}^{\left|\pi\right|} d(x_{\pi_1(i)}, y_{\pi_2(i)})
  \]

\[
  d_{\text{DTW}}(x, y) = \min_{\pi \in \text{Alignments}} c(\pi)
\]
Example: Optimal alignment between two strings

\[
d_{\text{DTW}}(x, y) = \min_{\pi \in \text{Alignments}} c(\pi)
\]

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

\[
k_{\text{GA}}(x, y) = \sum_{\pi \in \text{Alignments}} e^{-c(\pi)}
\]

is positive definite (C. et al. 2007, C. 2011)
**Example: Edit-distance between two trees**

- **Input**: two labeled trees $x, y$.

- **Mapping variable**: sequence of substitutions/deletions/insertions of vertices

- **Cost parameter**: $\gamma$ distance between labels and cost for deletion/insertion

\[
d_{\text{TreeEdit}}(x, y) = \min_{\sigma \in \text{EditScripts}(x, y)} \sum \gamma(\sigma_i)
\]
Example: Edit-distance between two trees

- **Input**: two labeled trees \( x, y \).

- **Mapping variable**: sequence of substitutions/deletions/insertions of vertices

\[
\text{TreeEdit}(x, y) = \min_{\sigma \in \text{EditScripts}(x, y)} \sum \gamma(\sigma_i)
\]

- **Cost parameter**: \( \gamma \) distance between labels and cost for deletion/insertion

- 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 \( x, y \in \mathbb{N}^d \) such that \( \sum_{i=1}^{d} x_i = \sum_{i=1}^{d} y_i = N \)

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

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

\[
d_W(x, y) = \min_{X \in U(r,c)} \langle X, M \rangle
\]
Example: Transportation distance between discrete histograms

\[ d_W(x, y) = \min_{X \in U(r,c)} \langle X, M \rangle \]

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

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

is positive definite (C., submitted).
To wrap up

\[ d(x, y) = \min_{\tau \in T(x,y)} c(\tau), \quad \delta(x, y) = \text{soft-min}_{\tau \in T(x,y)} c(\tau) \]

\[ e^{-\delta(x,y)} = \sum_{\tau \in T(x,y)} e^{-\langle \tau, \theta \rangle} = G_T(x,y)(\theta) \text{ is positive definite in many (all) cases.} \]
Open problems

• ∃ 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 $e^t$) (Shin 2011).

• To speed up computations, possible to restrict the sum to subset of $T(x, y)$?
  ○ C. 2011 with DTW.
  ○ C. submitted with transportation distances