Kernel Machines

Alain Celisse

SAMM

Paris 1-Panthéon Sorbonne University

alain.celisse@univ-paris1.fr

Lecture 4: Designing reproducing kernels

—

Master 2 Data Science – Centrale Lille, Lille University
Fall 2022

Outline of the lectures

Successive topics of the coming lectures:

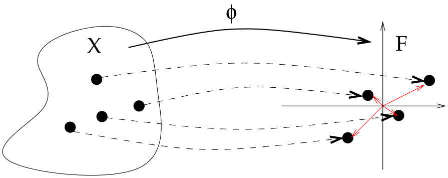
1. Introduction to Kernel methods
2. Support vector classifiers and Kernel methods
3. Extending classical strategies to high dimension
   - KRR/LS-SVMs
   - KPCA
4. Duality gap and KKT conditions
5. Designing reproducing kernels (Today!)
6. Maximum Mean Discrepancy (MMD)

Outline of the lecture

- PSD kernels
- Mercer's Theorem
- Designing PSD kernels
- Similarity measure
- Spectral Clustering

PSD kernels

PSD kernels: Motivation



- No vector-space structure on $\mathcal{X}$: computing a distance between observations is difficult
- Using PSD kernels: new representation of data as elements of a Hilbert space

Reminder

- **Inner-product:** Symmetric bilinear form on a vector space F such that $\langle x, x \rangle_F > 0$, for $x \in F \setminus \{0\}$
- **Pre-Hilbertian vector space:** Vector space endowed with an inner product
- **Hilbert space:** Pre-Hilbertian vector space which is complete

PSD kernel

- $\mathcal{X}$: a set (not necessarily a vector space)

Definition (Psd kernel)

A (real-valued) kernel $k(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semi-definite if

- $k(x,y) = k(y,x)$, for all $x, y \in \mathcal{X}$ (symmetric)
- $\forall n \in \mathbb{N}^*, \forall x_1, \ldots, x_n \in \mathcal{X}, \forall a_1, \ldots, a_n \in \mathbb{R}$,

$$\sum_{1 \leq i,j \leq n} a_i a_j k(x_i, x_j) \geq 0$$

which is equivalent to
$\forall n \in \mathbb{N}^*, \forall x_1, \ldots, x_n \in \mathcal{X}$,

$$\text{the matrix } \mathrm{K} = \{k(x_i, x_j)\}_{1 \leq i,j \leq n} \in \mathcal{S}_n^+(\mathbb{R})$$

(symmetric positive semi-definite matrices)

Basic examples of psd kernels (Cont'd)

Classical examples

- Linear kernel:

$$k(x, y) = \langle x, y \rangle_{\mathbb{R}^d}$$

- Polynomial kernel: $(c \geq 0,\ d > 0)$

$$k(x, y) = \left(\langle x, y \rangle_{\mathbb{R}^d} + c\right)^d$$

- Gaussian (Radial Basis Function) kernel:

$$k(x, y) = e^{-\frac{(x-y)^2}{2}}$$

Basic examples of psd kernels (Cont'd)

Classical examples

- Linear kernel:

$$k(x,y) = \langle x, y \rangle_{\mathbb{R}^d}$$

- Polynomial kernel: $(c \geq 0,\ d > 0)$

$$k(x,y) = \left(\langle x, y \rangle_{\mathbb{R}^d} + c\right)^d$$

- Gaussian (Radial Basis Function) kernel:

$$k(x,y) = e^{-\frac{(x-y)^2}{2}}$$

Remark:
The following reviews key properties which justify the claim that these functions are psd kernels

Basic examples of psd kernels (Cont'd)

General construction

- $\mathcal{H}$: pre-Hilbertian space endowed with $\langle \cdot, \cdot \rangle_{\mathcal{H}}$
- ϕ: $\mathcal{X} \to \mathcal{H}$: any mapping

Then, the kernel k defined by

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}\,, \qquad \forall x, y \in \mathcal{X}$$

is a psd kernel

Proof.
Just do it! $\square$

Basic examples of psd kernels (Cont'd)

General construction

- $\mathcal{H}$: pre-Hilbertian space endowed with $\langle \cdot, \cdot \rangle_{\mathcal{H}}$
- ϕ: $\mathcal{X} \to \mathcal{H}$: any mapping

Then, the kernel k defined by

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}, \qquad \forall x, y \in \mathcal{X}$$

is a psd kernel

Proof.
Just do it! $\square$

Example

With $\phi : \mathbb{R}^d \to \mathbb{R}$: coordinate-wise projection such that $\phi(x) = x_i$, then

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = x_i \cdot y_i$$

is a psd kernel

Motivation for Mercer's theorem

Previous result:

$$\text{Any mapping } \phi \Rightarrow \text{a psd kernel } k$$

Motivation for Mercer's theorem

Previous result:

$$\text{Any mapping } \phi \Rightarrow \text{a psd kernel } k$$

Key question

Does the reciprocal hold true?

that is, does it exist a mapping ϕ such that any psd kernel satisfies the above equation?

Motivation for Mercer's theorem

Previous result:

$$\text{Any mapping } \phi \Rightarrow \text{a psd kernel } k$$

Key question

Does the reciprocal hold true?

that is, does it exist a mapping ϕ such that any psd kernel satisfies the above equation?

$\longrightarrow$ This is where Mercer kernels come into play

(see also Steinwart and Christmann, 2008)

Kernel operator

- $(\mathcal{X}, \mathcal{B})$: measurable set (Borelian $\sigma-$algebra)
- ν: Borelian (finite) measure on $\mathcal{B}$
- $L_2(\mathcal{X}, \nu)$: space of square-integrable functions on $\mathcal{X}$

Definition (Kernel operator)

k: psd kernel such that $\int_{\mathcal{X}} k(x,x)d\nu(x) < +\infty$.
The mapping T_k defined, for $f \in L_2(\mathcal{X}, \nu)$, by

$$x \in \mathcal{X} \mapsto T_k(f)(x) = \int_{\mathcal{X}} k(x,y)f(y)d\nu(y)$$

is a linear operator from $L_2(\mathcal{X}, \nu) \to L_2(\mathcal{X}, \nu)$.
It is called the kernel operator associated with k.

Proof.

Defined on $L_2(\mathcal{X}, \nu)$, linear, $T_k(f) \in L_2(\mathcal{X}, \nu)$ $\square$

Remark:
Other assumptions on k are possible (e.g. Mercer kernels)

Proof hint

Proof.

- $$\begin{aligned} 0 &\leq \int_{\mathcal{X}} T_k(f)^2(x) d\nu(x) \\ &\leq \int_{\mathcal{X}} \left[\int_{\mathcal{X}} \left(k(x,t)\right)^2 d\nu(t) \cdot \int_{\mathcal{X}} f^2(t) d\nu(t) \right] d\nu(x) \\ &\leq \|f\|_{L_2}^2 \int_{\mathcal{X}} k(x,x) d\nu(x) \cdot \int_{\mathcal{X}} k(t,t) d\nu(t) \\ &\leq \|f\|_{L_2}^2 \left[\int_{\mathcal{X}} k(x,x) d\nu(x) \right]^2 < +\infty \end{aligned}$$

□

Mercer kernel

Definition (Mercer kernel)

- $\mathcal{X}$: metric space that is compact
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a continuous kernel

Psd kernel satisfying these conditions called Mercer kernel

Mercer kernel

Definition (Mercer kernel)

- $\mathcal{X}$: metric space that is compact
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a continuous kernel

Psd kernel satisfying these conditions called Mercer kernel

Remark:

With a Mercer kernel, $T_k : L_2(\mathcal{X}, \nu) \to L_2(\mathcal{X}, \nu)$ well defined.

Mercer kernel

Definition (Mercer kernel)

- $\mathcal{X}$: metric space that is compact
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a continuous kernel

Psd kernel satisfying these conditions called Mercer kernel

Remark:
With a Mercer kernel, $T_k : L_2(\mathcal{X}, \nu) \to L_2(\mathcal{X}, \nu)$ well defined.

Proof.

$$\begin{aligned}
&|T_k(f)(x) - T_k(f)(y)| \\
&\leq \|k(x,\cdot) - k(y,\cdot)\|_\nu \, \|f\|_\nu \\
&\leq \sup_{u \in \mathcal{X}} \|k(x,u) - k(y,u)\|_\infty \sqrt{\nu(\mathcal{X})} \, \|f\|_\nu
\end{aligned}$$

- $u \mapsto k(x,u) - k(y,u)$: Unif. cont. gives $T_k(f) \in C(\mathcal{X})$
- $\mathcal{X}$ compact implies $T_k(f) \in L_2(\mathcal{X}, \nu)$

□

From Mercer kernels to Spectral theorem

Theorem (Mercer kernel and Kernel operator)

For any Mercer kernel, $T_k : L_2(\mathcal{X}, \nu) \to L_2(\mathcal{X}, \nu)$ is semi-positive, self-adjoint, bounded and compact.

From Mercer kernels to Spectral theorem

Theorem (Mercer kernel and Kernel operator)

For any Mercer kernel, $T_k : L_2(\mathcal{X}, \nu) \to L_2(\mathcal{X}, \nu)$ is semi-positive, self-adjoint, bounded and compact.

From the spectral theorem applied to a linear compact operator on a Hilbert space:

Corollary

There exist:

- *Nonincreasing sequ. $\lambda_1 \geq \lambda_2 \geq \cdots > 0$ converging to 0*
- *Orthonormal family $\{\psi_i\}_{i\geq 1}$ of $L_2(\mathcal{X}, \nu)$ such that*

$$\forall f \in L_2(\mathcal{X}, \nu), \qquad T_k(f) = \sum_{i\geq 1} \lambda_i \left\langle f, \psi_i \right\rangle_{L_2} \psi_i$$

From Mercer kernels to Spectral theorem

Theorem (Mercer kernel and Kernel operator)

For any Mercer kernel, $T_k : L_2(\mathcal{X}, \nu) \to L_2(\mathcal{X}, \nu)$ is semi-positive, self-adjoint, bounded and compact.

From the spectral theorem applied to a linear compact operator on a Hilbert space:

Corollary

There exist:

- *Nonincreasing sequ. $\lambda_1 \geq \lambda_2 \geq \cdots > 0$ converging to 0*
- *Orthonormal family $\{\psi_i\}_{i\geq 1}$ of $L_2(\mathcal{X}, \nu)$ such that*

$$\forall f \in L_2(\mathcal{X}, \nu), \qquad T_k(f) = \sum_{i\geq 1} \lambda_i \left\langle f, \psi_i \right\rangle_{L_2} \psi_i$$

Remark:

- $\lambda_1 \geq \lambda_2 \geq \cdots > 0$: Eigenvalues of T_k
- $\{\psi_i\}_{i\geq 1}$: Eigenvectors of T_k

Mercer's theorem

Theorem (Mercer's theorem)

For any Mercer kernel on $(\mathcal{X}, \nu)$*, there exist a nonincreasing sequence* $\lambda_1 \geq \lambda_2 \geq \cdots > 0$ *and an orthonormal family* $\{\psi_i\}_{i \geq 1}$ *of* $L_2(\mathcal{X}, \nu)$ *such that*

$$k(x, y) = \sum_{i \geq 1} \lambda_i \psi_i(x) \psi_i(y), \qquad \forall x, y \in \mathcal{X}$$

where the sum is absolutely convergent for each $(x, y) \in \mathcal{X}^2$

Mercer's theorem

Theorem (Mercer's theorem)

For any Mercer kernel on $(\mathcal{X}, \nu)$*, there exist a nonincreasing sequence* $\lambda_1 \geq \lambda_2 \geq \cdots > 0$ *and an orthonormal family* $\{\psi_i\}_{i\geq 1}$ *of* $L_2(\mathcal{X}, \nu)$ *such that*

$$k(x, y) = \sum_{i\geq 1} \lambda_i \psi_i(x)\psi_i(y), \qquad \forall x, y \in \mathcal{X}$$

where the sum is absolutely convergent for each $(x, y) \in \mathcal{X}^2$

Remark:

With $x = y$, $k(x, x) = \sum_{i=1}^{n} \lambda_i \psi_i^2(x) < +\infty$ implies that $\left\{\sqrt{\lambda_i \psi_i(x)}\right\}_{i\geq 1} \in \ell_2(\mathbb{R})$ for every x

Mercer and sdp kernels

Corollary (Mapping ϕ)

For any Mercer kernel on $(\mathcal{X}, \nu)$

$$\phi : \quad x \in \mathcal{X} \mapsto \phi(x) = \left\{ \sqrt{\lambda_i \psi_i(x)} \right\}_{i \geq 1} \in \ell_2(\mathbb{R})$$

is well defined, continuous, and satisfies

$$\forall x, y \in \mathcal{X}, \qquad k(x, y) = \langle \phi(x), \phi(u) \rangle_{\ell_2(\mathbb{R})}$$

Mercer and sdp kernels

Corollary (Mapping ϕ)

For any Mercer kernel on $(\mathcal{X}, \nu)$

$$\phi : \quad x \in \mathcal{X} \mapsto \phi(x) = \left\{ \sqrt{\lambda_i \psi_i(x)} \right\}_{i \geq 1} \in \ell_2(\mathbb{R})$$

is well defined, continuous, and satisfies

$$\forall x, y \in \mathcal{X}, \qquad k(x,y) = \langle \phi(x), \phi(u) \rangle_{\ell_2(\mathbb{R})}$$

Proof.

$$\begin{aligned} \|\phi(x) - \phi(y)\|^2_{\ell_2(\mathbb{R})} &= \sum_{i \geq 1} \lambda_i \left(\phi_i(x) - \phi_i(y) \right)^2 \\ &= k(x,x) - 2k(x,y) + k(y,y) \end{aligned}$$

$\square$

Mercer and sdp kernels

Corollary (Mapping ϕ)

For any Mercer kernel on $(\mathcal{X}, \nu)$

$$\phi : \quad x \in \mathcal{X} \mapsto \phi(x) = \left\{ \sqrt{\lambda_i \psi_i(x)} \right\}_{i \geq 1} \in \ell_2(\mathbb{R})$$

is well defined, continuous, and satisfies

$$\forall x, y \in \mathcal{X}, \qquad k(x, y) = \langle \phi(x), \phi(u) \rangle_{\ell_2(\mathbb{R})}$$

Proof.

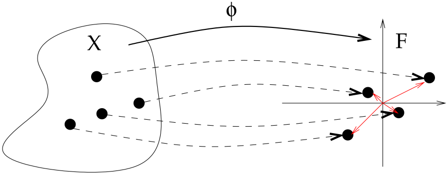
$$\begin{aligned} \|\phi(x) - \phi(y)\|^2_{\ell_2(\mathbb{R})} &= \sum_{i \geq 1} \lambda_i \left(\phi_i(x) - \phi_i(y) \right)^2 \\ &= k(x, x) - 2k(x, y) + k(y, y) \end{aligned}$$

$\square$

Remarks:

- Any Mercer kernel $\Rightarrow$ a mapping ϕ
- Similar result with translation-invariant kernels

Psd kernels and Distance



Distance between x and y in $\mathcal{X}$

- $x, y \in \mathcal{X}$
- k: psd kernel and ϕ as above
- $\phi(x), \phi(y) \in \mathcal{H}$: pre-Hilbertian space

$$\begin{aligned} d(x,y)^2 &= \|\phi(x) - \phi(y)\|_{\mathcal{H}}^2 \\ &= k(x,x) - 2k(x,y) + k(y,y) \end{aligned}$$

(kernel trick)

Designing PSD kernels: Classical rules

Nonnegative sum, product

- k_1, k_2: psd kernels on $\mathcal{X}$

Proposition

- **Nonnegative sum:** $\alpha_1 k_1 + \alpha_2 k_2$ *is a psd kernel* $(\alpha_1, \alpha_2 \geq 0)$

Nonnegative sum, product

- k_1, k_2: psd kernels on $\mathcal{X}$

Proposition

- **Nonnegative sum:** *$\alpha_1 k_1 + \alpha_2 k_2$ is a psd kernel ($\alpha_1, \alpha_2 \geq 0$)*
- **Product:** *The kernel $k_1 k_2$ on $\mathcal{X}$ given by*

$$k_1 k_2(x, y) = k_1(x, y) \cdot k_2(x, y)$$

is a psd kernel

Nonnegative sum, product

- k_1, k_2: psd kernels on $\mathcal{X}$

Proposition

- **Nonnegative sum:** $\alpha_1 k_1 + \alpha_2 k_2$ *is a psd kernel* $(\alpha_1, \alpha_2 \geq 0)$
- **Product:** *The kernel* $k_1 k_2$ *on* $\mathcal{X}$ *given by*

$$k_1 k_2(x, y) = k_1(x, y) \cdot k_2(x, y)$$

is a psd kernel

Example

- **Polynomial kernel:** psd (Nonneg. constant kernel is psd)

Nonnegative sum, product

- k_1, k_2: psd kernels on $\mathcal{X}$

Proposition

- **Nonnegative sum:** $\alpha_1 k_1 + \alpha_2 k_2$ *is a psd kernel* $(\alpha_1, \alpha_2 \geq 0)$
- **Product:** *The kernel* $k_1 k_2$ *on* $\mathcal{X}$ *given by*

$$k_1 k_2(x, y) = k_1(x, y) \cdot k_2(x, y)$$

is a psd kernel

Example

- **Polynomial kernel:** psd (Nonneg. constant kernel is psd)
- **Conformal transformation:**
  With $k(x, y)$ psd, and $k'(x, y) = f(x)f(y)$ (where $f \geq 0$)
  Then $k_f(x, y) = f(x)k(x, y)f(y)$ is psd

Normalized kernel

Normalized kernel

If k is psd on $\mathcal{X}$, then the kernel $\widetilde{k}$ given by

$$\widetilde{k}(x,y) = \frac{k(x,y)}{\sqrt{k(x,x)} \cdot \sqrt{k(y,y)}} \cdot \mathbb{1}_{k(x,x)\cdot k(y,y)>0}, \qquad \forall x,y \in, \mathcal{X}$$

is a psd kernel

Proof.

Do it yourself! $\square$

Example: Cosinus

Cosinus interpretation

If there exist ϕ and a pre-Hilbertian space $\mathcal{H}$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$, then

$$\widetilde{k}(x, y) = \frac{\langle \phi(x), \phi(y) \rangle_{\mathcal{H}}}{\|\phi(x)\|_{\mathcal{H}} \cdot \|\phi(y)\|_{\mathcal{H}}} \in [-1, 1], \qquad \forall x, y \in, \mathcal{X}$$

is the cosinus of the angle between $\phi(x)$ and $\phi(y)$ within $\mathcal{H}$

Example: Cosinus

Cosinus interpretation

If there exist ϕ and a pre-Hilbertian space $\mathcal{H}$ such that $k(x,y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$, then

$$\widetilde{k}(x,y) = \frac{\langle \phi(x), \phi(y) \rangle_{\mathcal{H}}}{\|\phi(x)\|_{\mathcal{H}} \cdot \|\phi(y)\|_{\mathcal{H}}} \in [-1,1], \qquad \forall x,y \in, \mathcal{X}$$

is the cosinus of the angle between $\phi(x)$ and $\phi(y)$ within $\mathcal{H}$

Limit and Power series of inner-product

- $k_1, k_2, \ldots, k_p, \ldots$: sequence of psd kernels on $\mathcal{X}$

Proposition

- *Limit: Any kernel k (well) defined by* $\lim_{p \to +\infty} k_p(x, y) = k(x, y)$ *for every* $x, y \in \mathcal{X}$ *is a psd kernel*

Limit and Power series of inner-product

- $k_1, k_2, \ldots, k_p, \ldots$: sequence of psd kernels on $\mathcal{X}$

Proposition

- *Limit: Any kernel k (well) defined by* $\lim_{p \to +\infty} k_p(x, y) = k(x, y)$ *for every* $x, y \in \mathcal{X}$ *is a psd kernel*
- *Power series of inner-product:* *With* $h(t) = \sum_{i \geq 1} a_i t^i$ *(defined on* $\mathbb{R}$*), the kernel* $k(x, y) = h(\langle x, y \rangle)$ *is psd iff the sequence* $\{a_i\}_{i \geq 1}$ *is nonnegative*

Limit and Power series of inner-product

- $k_1, k_2, \ldots, k_p, \ldots$: sequence of psd kernels on $\mathcal{X}$

Proposition

- *Limit: Any kernel k (well) defined by* $\lim_{p\to+\infty} k_p(x,y) = k(x,y)$ *for every* $x, y \in \mathcal{X}$ *is a psd kernel*
- *Power series of inner-product:* *With* $h(t) = \sum_{i\geq 1} a_i t^i$ *(defined on* $\mathbb{R}$*), the kernel* $k(x,y) = h(\langle x, y\rangle)$ *is psd iff the sequence* $\{a_i\}_{i\geq 1}$ *is nonnegative*

Example

The exponential kernel given by $k(x,y) = \exp(\langle x, y\rangle)$ is psd

Exercise with the Gaussian kernel

Prove that the Gaussian kernel is psd

Exercise with the Gaussian kernel

Prove that the Gaussian kernel is psd

Hint:
Factorize the Gaussian kernel and use the conformal transformation

Exercise with the Gaussian kernel

Prove that the Gaussian kernel is psd

Hint:
Factorize the Gaussian kernel and use the conformal transformation

$$
\begin{aligned}
e^{\frac{1}{h}\langle x,y\rangle} &= e^{\frac{1}{2h}\|x\|^2} \cdot e^{-\frac{1}{2h}\|x-y\|^2} \cdot e^{\frac{1}{2h}\|y\|^2} \\
&= \frac{e^{-\frac{1}{2h}\|x-y\|^2}}{e^{-\frac{1}{2h}\|x\|^2} \cdot e^{-\frac{1}{2h}\|y\|^2}}
\end{aligned}
$$

Translation invariant

Definition (Translation-invariant kernel)

Any kernel k defined on $\mathcal{X}$ by $k(x, y) = h(x - y)$ is a **translation-invariant** kernel.

Translation invariant

Definition (Translation-invariant kernel)

Any kernel k defined on $\mathcal{X}$ by $k(x, y) = h(x - y)$ is a **translation-invariant** kernel.

Proposition

Any translation invariant kernel k is psd on $\mathcal{X} \subset \mathbb{R}^d$ if the Fourier transform of h is nonnegative that is,

$$\mathcal{F}(h)(w) = \frac{1}{(2\pi)^{d/2}} \int_{\mathcal{X}} e^{-i\langle w,x\rangle} h(x) dx \geq 0$$

Translation invariant

Definition (Translation-invariant kernel)

Any kernel k defined on $\mathcal{X}$ by $k(x, y) = h(x - y)$ is a **translation-invariant** kernel.

Proposition

Any translation invariant kernel k is psd on $\mathcal{X} \subset \mathbb{R}^d$ if the Fourier transform of h is nonnegative that is,

$$\mathcal{F}(h)(w) = \frac{1}{(2\pi)^{d/2}} \int_{\mathcal{X}} e^{-i\langle w,x \rangle} h(x)dx \geq 0$$

Remark:

This is a sufficient condition (which is not necessary)

Motivation: Structured objects

Fact: In practice, an individual is often described by "different types" of features:

- **Qualitative data:** hair color, gender, nationality,...

Motivation: Structured objects

Fact: In practice, an individual is often described by "different types" of features:

- **Qualitative data:** hair color, gender, nationality,...
- **Quantitative data:** age, blood pressure, temperature

Motivation: Structured objects

Fact: In practice, an individual is often described by "different types" of features:

- **Qualitative data:** hair color, gender, nationality,...
- **Quantitative data:** age, blood pressure, temperature
- **Ranked data:** Sequece of objects ordered according to some visbility, preference, priority criteria

Motivation: Structured objects

Fact: In practice, an individual is often described by "different types" of features:

- **Qualitative data:** hair color, gender, nationality,...
- **Quantitative data:** age, blood pressure, temperature
- **Ranked data:** Sequece of objects ordered according to some visbility, preference, priority criteria
- Social networks, time series,...

Motivation: Structured objects

Fact: In practice, an individual is often described by "different types" of features:

- **Qualitative data:** hair color, gender, nationality,...
- **Quantitative data:** age, blood pressure, temperature
- **Ranked data:** Sequece of objects ordered according to some visbility, preference, priority criteria
- Social networks, time series,...

Classical challenge

- Combining all these types of descriptors is highly challenging with classical strategies

Motivation: Structured objects

Fact: In practice, an individual is often described by "different types" of features:

- **Qualitative data:** hair color, gender, nationality,. . .
- **Quantitative data:** age, blood pressure, temperature
- **Ranked data:** Sequece of objects ordered according to some visbility, preference, priority criteria
- Social networks, time series,. . .

Classical challenge

- Combining all these types of descriptors is highly challenging with classical strategies

- **Kernel-based strategy:**
  1. Design one kernel for each type of feature

Motivation: Structured objects

Fact: In practice, an individual is often described by "different types" of features:

- **Qualitative data:** hair color, gender, nationality,...
- **Quantitative data:** age, blood pressure, temperature
- **Ranked data:** Sequece of objects ordered according to some visbility, preference, priority criteria
- Social networks, time series,...

Classical challenge

- Combining all these types of descriptors is highly challenging with classical strategies

- **Kernel-based strategy:**
  1. Design one kernel for each type of feature
  2. Design one new kernel "combining" each type of data

Tensor product and direct sum

Structured objects

- $x_1, y_1 \in \mathcal{X}_1$: first feature type
- $x_2, y_2 \in \mathcal{X}_2$: second feature type
- $x = (x_1, x_2)$ and $y = (y_1, y_2)$
- k_1 psd kernel based on $\mathcal{X}_1$
- k_2 psd kernel based on $\mathcal{X}_2$

Tensor product and direct sum

Structured objects

- $x_1, y_1 \in \mathcal{X}_1$: first feature type
- $x_2, y_2 \in \mathcal{X}_2$: second feature type
- $x = (x_1, x_2)$ and $y = (y_1, y_2)$
- k_1 psd kernel based on $\mathcal{X}_1$
- k_2 psd kernel based on $\mathcal{X}_2$

Tensor product and Direct sum

The tensor product (resp. direct sum) kernel is defined by

$$(k_1 \otimes k_2)(x, y) = k_1(x_1, y_1) \cdot k_2(x_2, y_2)$$
$$(k_1 \oplus k_2)(x, y) = k_1(x_1, y_1) + k_2(x_2, y_2)$$

for all $x = (x_1, x_2), y = (y_1, y_2) \in \mathcal{X}_1 \times \mathcal{X}_2$

ANOVA kernel

- L types/levels of features
- $\mathcal{X} = \prod_{\ell=1}^{L} \mathcal{X}_\ell$
- For each ℓ, k_ℓ psd kernel focusing on $\mathcal{X}_\ell$
- D: Order of interaction (number of interacting levels)

ANOVA kernel

- L types/levels of features
- $\mathcal{X} = \prod_{\ell=1}^{L} \mathcal{X}_\ell$
- For each ℓ, k_ℓ psd kernel focusing on $\mathcal{X}_\ell$
- D: Order of interaction (number of interacting levels)

Interacting levels

The interaction between two levels ℓ, ℓ' is accounted for if the tensor product kernel $k_\ell \otimes k_{\ell'}$ is considered

ANOVA kernel

- L types/levels of features
- $\mathcal{X} = \prod_{\ell=1}^{L} \mathcal{X}_\ell$
- For each ℓ, k_ℓ psd kernel focusing on $\mathcal{X}_\ell$
- D: Order of interaction (number of interacting levels)

Interacting levels

The interaction between two levels ℓ, ℓ' is accounted for if the tensor product kernel $k_\ell \otimes k_{\ell'}$ is considered

ANOVA kernel of order D $(1 \leq D \leq L)$

$$k_D(x, y) = \sum_{1 \leq \ell_1 < \cdots < \ell_D \leq L} \left[\prod_{i=1}^{D} k_{\ell_i}\left(x_{\ell_i}, y_{\ell_i}\right) \right]$$

ANOVA kernel

- L types/levels of features
- $\mathcal{X} = \prod_{\ell=1}^{L} \mathcal{X}_\ell$
- For each ℓ, k_ℓ psd kernel focusing on $\mathcal{X}_\ell$
- D: Order of interaction (number of interacting levels)

Interacting levels

The interaction between two levels ℓ, ℓ' is accounted for if the tensor product kernel $k_\ell \otimes k_{\ell'}$ is considered

ANOVA kernel of order D $(1 \leq D \leq L)$

$$k_D(x, y) = \sum_{1 \leq \ell_1 < \cdots < \ell_D \leq L} \left[\prod_{i=1}^{D} k_{\ell_i}\left(x_{\ell_i}, y_{\ell_i}\right) \right]$$

Remark:

- $D = 1$: direct sum kernel

ANOVA kernel

- L types/levels of features
- $\mathcal{X} = \prod_{\ell=1}^{L} \mathcal{X}_\ell$
- For each ℓ, k_ℓ psd kernel focusing on $\mathcal{X}_\ell$
- D: Order of interaction (number of interacting levels)

Interacting levels

The interaction between two levels ℓ, ℓ' is accounted for if the tensor product kernel $k_\ell \otimes k_{\ell'}$ is considered

ANOVA kernel of order D $(1 \leq D \leq L)$

$$k_D(x, y) = \sum_{1 \leq \ell_1 < \cdots < \ell_D \leq L} \left[\prod_{i=1}^{D} k_{\ell_i}\left(x_{\ell_i}, y_{\ell_i}\right) \right]$$

Remark:

- $D = 1$: direct sum kernel
- $D = L$: tensor product kernel

ANOVA kernel

- L types/levels of features
- $\mathcal{X} = \prod_{\ell=1}^{L} \mathcal{X}_\ell$
- For each ℓ, k_ℓ psd kernel focusing on $\mathcal{X}_\ell$
- D: Order of interaction (number of interacting levels)

Interacting levels

The interaction between two levels ℓ, ℓ' is accounted for if the tensor product kernel $k_\ell \otimes k_{\ell'}$ is considered

ANOVA kernel of order D $(1 \leq D \leq L)$

$$k_D(x, y) = \sum_{1 \leq \ell_1 < \cdots < \ell_D \leq L} \left[\prod_{i=1}^{D} k_{\ell_i}\left(x_{\ell_i}, y_{\ell_i}\right) \right]$$

Remark:

- $D = 1$: direct sum kernel
- $D = L$: tensor product kernel
- $1 \leq D \leq L$: ANOVA between the direct sum (no interaction) and the tensor (full interaction) product

Similarity measure: Examples

Similarities between (finite) sets

- $\mathcal{C}$: collection of d (sub)sets $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of a set $\mathcal{S}$
  Ex:
  $\mathcal{S} = \{1, 2, 3\}$, $\mathcal{X}_1 = \{1\}$, $\mathcal{X}_2 = \{1, 2\}$, $\mathcal{X}_3 = \{1, 3\}$
  $\longrightarrow \mathcal{C} = \{\{1\}, \{1, 2\}, \{1, 3\}\}$ that is, $d = 3$

Similarities between (finite) sets

- $\mathcal{C}$: collection of d (sub)sets $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of a set $\mathcal{S}$
  Ex:
  $\mathcal{S} = \{1,2,3\}$, $\mathcal{X}_1 = \{1\}$, $\mathcal{X}_2 = \{1,2\}$, $\mathcal{X}_3 = \{1,3\}$
  $\longrightarrow \mathcal{C} = \{\{1\},\{1,2\},\{1,3\}\}$ that is, $d = 3$
- μ: nonnegative measure on $\mathcal{P}(\mathcal{S})$

Similarities between (finite) sets

- $\mathcal{C}$: collection of d (sub)sets $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of a set $\mathcal{S}$
  Ex:
  $\mathcal{S} = \{1,2,3\}$, $\mathcal{X}_1 = \{1\}$, $\mathcal{X}_2 = \{1,2\}$, $\mathcal{X}_3 = \{1,3\}$
  $\longrightarrow \mathcal{C} = \{\{1\}, \{1,2\}, \{1,3\}\}$ that is, $d = 3$
- μ: nonnegative measure on $\mathcal{P}(\mathcal{S})$

Intersection kernel

$$k(\mathcal{X}, \mathcal{Y}) = \mu\left(\mathcal{X} \cap \mathcal{Y}\right), \qquad \forall \mathcal{X}, \mathcal{Y} \in \mathcal{C}$$

Similarities between (finite) sets

- $\mathcal{C}$: collection of d (sub)sets $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of a set $\mathcal{S}$
  Ex:
  $\mathcal{S} = \{1, 2, 3\}$, $\mathcal{X}_1 = \{1\}$, $\mathcal{X}_2 = \{1, 2\}$, $\mathcal{X}_3 = \{1, 3\}$
  $\longrightarrow \mathcal{C} = \{\{1\}, \{1, 2\}, \{1, 3\}\}$ that is, $d = 3$
- μ: nonnegative measure on $\mathcal{P}(\mathcal{S})$

Intersection kernel

$$k(\mathcal{X}, \mathcal{Y}) = \mu(\mathcal{X} \cap \mathcal{Y}), \qquad \forall \mathcal{X}, \mathcal{Y} \in \mathcal{C}$$

Remark:

Does not depend on the size of each of $\mathcal{X}$ and $\mathcal{Y}$

Similarities between (finite) sets

- $\mathcal{C}$: collection of d (sub)sets $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of a set $\mathcal{S}$
  Ex:
  $\mathcal{S} = \{1,2,3\}$, $\mathcal{X}_1 = \{1\}$, $\mathcal{X}_2 = \{1,2\}$, $\mathcal{X}_3 = \{1,3\}$
  $\longrightarrow \mathcal{C} = \{\{1\}, \{1,2\}, \{1,3\}\}$ that is, $d = 3$
- μ: nonnegative measure on $\mathcal{P}(\mathcal{S})$

Intersection kernel

$$k(\mathcal{X}, \mathcal{Y}) = \mu\left(\mathcal{X} \cap \mathcal{Y}\right), \qquad \forall \mathcal{X}, \mathcal{Y} \in \mathcal{C}$$

Remark:
Does not depend on the size of each of $\mathcal{X}$ and $\mathcal{Y}$

Normalized Intersection kernel

$$k(\mathcal{X}, \mathcal{Y}) = \frac{\mu\left(\mathcal{X} \cap \mathcal{Y}\right)}{\mu\left(\mathcal{X} \cup \mathcal{Y}\right)}, \qquad \forall \mathcal{X}, \mathcal{Y} \in \mathcal{C}$$

Similarities between (finite) sets

- $\mathcal{C}$: collection of d (sub)sets $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of a set $\mathcal{S}$
  Ex:
  $\mathcal{S} = \{1,2,3\}$, $\mathcal{X}_1 = \{1\}$, $\mathcal{X}_2 = \{1,2\}$, $\mathcal{X}_3 = \{1,3\}$
  $\longrightarrow \mathcal{C} = \{\{1\},\{1,2\},\{1,3\}\}$ that is, $d = 3$
- μ: nonnegative measure on $\mathcal{P}(\mathcal{S})$

Intersection kernel

$$k(\mathcal{X},\mathcal{Y}) = \mu\left(\mathcal{X} \cap \mathcal{Y}\right), \qquad \forall \mathcal{X},\mathcal{Y} \in \mathcal{C}$$

Remark:
Does not depend on the size of each of $\mathcal{X}$ and $\mathcal{Y}$

Normalized Intersection kernel

$$k(\mathcal{X},\mathcal{Y}) = \frac{\mu\left(\mathcal{X} \cap \mathcal{Y}\right)}{\mu\left(\mathcal{X} \cup \mathcal{Y}\right)}, \qquad \forall \mathcal{X},\mathcal{Y} \in \mathcal{C}$$

Remark:
Comparison of clusterings, graphs, vectors of categorical data

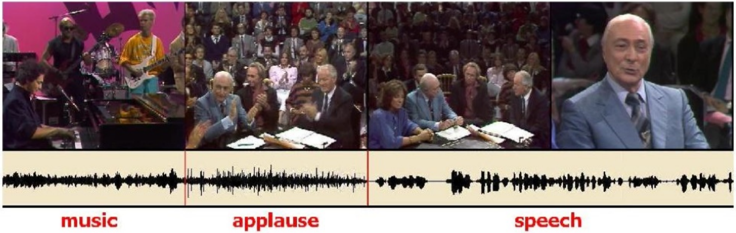
Similarities between sets

- $\mathcal{C}$: σ-algebra
- μ: nonnegative measure on $\mathcal{C}$ with $\mu(\mathcal{X}) < +\infty$
- $X, Y \subset \mathcal{X}$

Intersection kernel For all $X, Y \in \mathcal{C}$,

$$k(X, Y) = \mu(X \cap Y) = \int_{\mathcal{X}} \mathbb{1}_X(u) \cdot \mathbb{1}_Y(u) d\mu(u)$$

Motivating example 1: Structured objects



Description:

- Video sequences from "Le grand échiquier", 70s-80s French talk show.
- At each time, one observes an image (high-dimensional).
- Each image is summarized by a histogram.

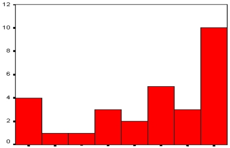
Motivating example 1: Structured objects

- Preprocessing images (patches in yellow).

- Each histogram bin corresponds to a patch.

Non-vectorial object:

Each image $\Rightarrow$



$\longrightarrow$ Algorithms for vectorial data are not accurate.

Histograms (Bag-of-words)

χ^2-distance between histograms

- $p = (p_1, \ldots, p_I)$: histogram with I bins $\quad (\sum_{i=1}^{I} p_i = 1)$
- $q = (q_1, \ldots, q_I)$: histogram with I bins

$$d_{\chi^2}(p, q) = \sum_{i=1}^{I} \frac{(p_i - q_i)^2}{p_i + q_i}$$

Histograms (Bag-of-words)

χ^2-distance between histograms

- $p = (p_1, \ldots, p_I)$: histogram with I bins $\quad (\sum_{i=1}^{I} p_i = 1)$
- $q = (q_1, \ldots, q_I)$: histogram with I bins

$$d_{\chi^2}(p, q) = \sum_{i=1}^{I} \frac{(p_i - q_i)^2}{p_i + q_i}$$

χ^2-kernel

The kernel defined by

$$k_{\chi^2, h}(p, q) = e^{-\frac{1}{h} d_{\chi^2}(p,q)}$$

is psd

Histograms (Bag-of-words)

χ^2-distance between histograms

- $p = (p_1, \ldots, p_I)$: histogram with I bins $\quad (\sum_{i=1}^{I} p_i = 1)$
- $q = (q_1, \ldots, q_I)$: histogram with I bins

$$d_{\chi^2}(p, q) = \sum_{i=1}^{I} \frac{(p_i - q_i)^2}{p_i + q_i}$$

χ^2-kernel

The kernel defined by

$$k_{\chi^2,h}(p, q) = e^{-\frac{1}{h} d_{\chi^2}(p,q)}$$

is psd

Example

Video streams, documents (counts of words), graphs (motif counts),...

Likelihoods

- $\{P_\theta \mid \theta \in \Theta\}$: statistical model, $\Theta \subset \mathbb{R}^d$
- Dominating measure μ with $P_\theta \ll mu$: $f_\theta = \frac{dP_\theta}{d\mu}$
- $X = (x_1, \ldots, x_m)$: m-samples
- $Y = (y_1, \ldots, y_n)$: n-samples

Kernel between score vectors

Score vectors:

- $\dot{\ell}_\theta(X) = \partial_\theta \log\left(f_\theta(X)\right) \in \mathbb{R}^d$
- $\dot{\ell}_\theta(Y) = \partial_\theta \log\left(f_\theta(Y)\right) \in \mathbb{R}^d$

Likelihoods

- $\{P_\theta \mid \theta \in \Theta\}$: statistical model, $\Theta \subset \mathbb{R}^d$
- Dominating measure μ with $P_\theta \ll mu$: $f_\theta = \frac{dP_\theta}{d\mu}$
- $X = (x_1, \ldots, x_m)$: m-samples
- $Y = (y_1, \ldots, y_n)$: n-samples

Kernel between score vectors

Score vectors:

- $\dot{\ell}_\theta(X) = \partial_\theta \log\left(f_\theta(X)\right) \in \mathbb{R}^d$
- $\dot{\ell}_\theta(Y) = \partial_\theta \log\left(f_\theta(Y)\right) \in \mathbb{R}^d$

$$k(X, Y) = \dot{\ell}_\theta(X)^\top \dot{\ell}_\theta(Y) = \left\langle \dot{\ell}_\theta(X), \dot{\ell}_\theta(Y) \right\rangle_{\mathbb{R}^d}$$

Likelihoods

- $\{P_\theta \mid \theta \in \Theta\}$: statistical model, $\Theta \subset \mathbb{R}^d$
- Dominating measure μ with $P_\theta \ll mu$: $f_\theta = \frac{dP_\theta}{d\mu}$
- $X = (x_1, \ldots, x_m)$: m-samples
- $Y = (y_1, \ldots, y_n)$: n-samples

Kernel between score vectors

Score vectors:

- $\dot{\ell}_\theta(X) = \partial_\theta \log\left(f_\theta(X)\right) \in \mathbb{R}^d$
- $\dot{\ell}_\theta(Y) = \partial_\theta \log\left(f_\theta(Y)\right) \in \mathbb{R}^d$

$$k(X, Y) = \dot{\ell}_\theta(X)^\top \dot{\ell}_\theta(Y) = \left\langle \dot{\ell}_\theta(X), \dot{\ell}_\theta(Y) \right\rangle_{\mathbb{R}^d}$$

Remark:

- Related to the Tangent Kernel (limit of infinite DNNs. . .)

Fisher's kernel

- Ficher information matrix:

$$I(\theta) = \mathbb{E}_\theta \left[\dot{\ell}_\theta(X) \cdot \dot{\ell}_\theta(X)^\top \right] \in \mathcal{M}_d(\mathbb{R})$$

- Fisher kernel between X and Y

$$k_{F,\theta}(X, Y) = \dot{\ell}_\theta(X)^\top \cdot I(\theta)^{-1} \cdot \dot{\ell}_\theta(Y)$$

Compare strings

Compare strings by means of substrings they contain

Notations:

- Σ: alphabet, Σ^n: language of words with length n
- $\Sigma^* = \cup_{n\geq 0}\Sigma^n$: language of words of any length
- $s \in \Sigma^*$
- $u \in \Sigma^n$ with $n \leq |s|$
- $\mathrm{i} = (i_1, \ldots, i_n)$ with $\ell(\mathrm{i}) = i_n - i_1 + 1$

Feature map

For any $u \in \Sigma^n$,

$$[\,\phi(s)\,]_u = \sum_{1\leq i_1<\cdots<i_n\leq|s|} \mathbb{1}_{(s(\mathrm{i})=u)} \lambda^{\ell(i)}$$

with $0 < \lambda \leq 1$

Compare strings (Cont'd)

For any $u \in \Sigma^n$,

$$[\phi(s)]_u = \sum_{1 \le i_1 < \cdots < i_n \le |s|} \mathbb{1}_{(s(\mathrm{i})=u)} \lambda^{\ell(i)}$$

with $0 < \lambda \le 1$

Example

- $s = \texttt{triangle rectan}$, $u = \texttt{gle}$
- $[\phi(s)]_u = \lambda^3 + \lambda^6$

String kernel

For all pairs of strings s, t,

$$k_n(s,t) = \sum_{|u|=n} [\phi(s)]_u \cdot [\phi(t)]_u$$

$$= \sum_{|u|=n} \left[\sum_{\mathrm{i,j} | s(\mathrm{i})=u=t(\mathrm{j})} \lambda^{\ell(\mathrm{i})} \cdot \lambda^{\ell(\mathrm{j})} \right]$$

Designing graph kernels

Deserves a whole class in its own!

Designing graph kernels

Deserves a whole class in its own!

$\longrightarrow$ See *Survey on Graph Kernels*
by Kriege, Johansson, and Morris (2020)

Spectral clustering

Psd kernel and similarity measure

Structured Observations

- $X_1, \ldots, X_n \in \mathcal{X}$: n-sample of observations
- $\mathcal{X}$: a general set (no vector space!)
- **In general:** difficult to tackle...

Psd kernel and similarity measure

Structured Observations

- $X_1, \ldots, X_n \in \mathcal{X}$: n-sample of observations
- $\mathcal{X}$: a general set (no vector space!)
- **In general:** difficult to tackle. . .

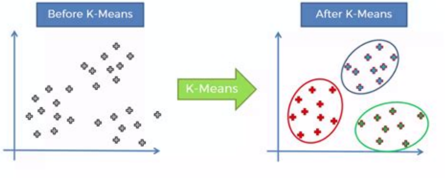
Designing a similarity measure

- For each type/level of features, design a psd kernel
- Use an ANOVA-like kernel k for combining these feature levels/types
- For each couple (X_i, X_j), similarity $S_{i,j}$ measured by means of

$$S_{i,j} = k(X_i, X_j), \qquad \forall 1 \leq i, j \leq n$$

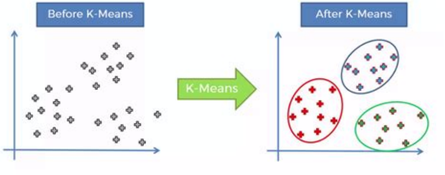
Motivation (1/2)

"Classical" clustering approaches

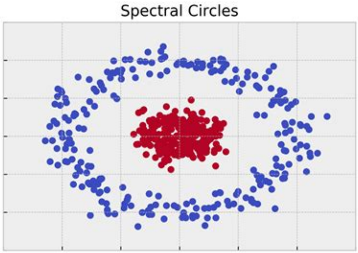


Motivation (1/2)

"Classical" clustering approaches

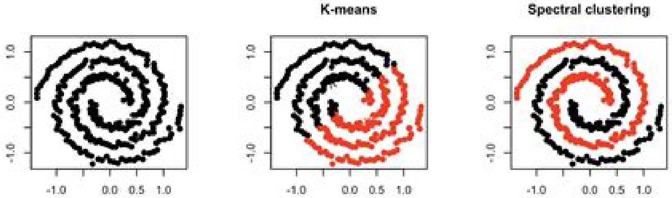


Challenging example for K-means



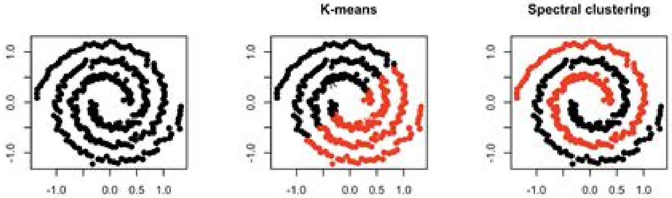
Motivation (2/2)

How spectral clustering improves upon K-means



Motivation (2/2)

How spectral clustering improves upon K-means

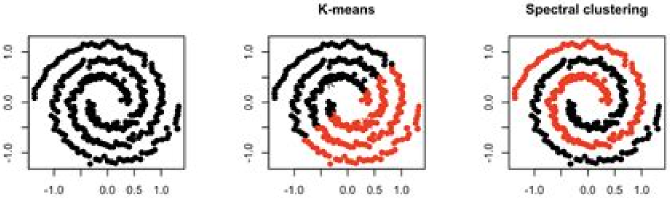


Based on two assumptions:

1. Several "*connected components*" do exist

Motivation (2/2)

How spectral clustering improves upon K-means



Based on two assumptions:

1. Several "*connected components*" do exist
2. "Distance" between neighbors within each *connected component* is small
   compared to
   that of neighbors between connected components

From neighbors to similarity graphs (1/2)

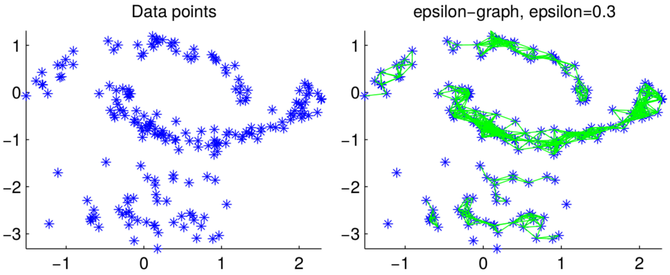
ϵ-neighborhood graph

- i and j connected if: $S_{i,j} \leq \epsilon$
- i and j connected: $W_{i,j} = 1$ ($W_{i,j} = 0$ otherwise)

From neighbors to similarity graphs (1/2)

ϵ-neighborhood graph

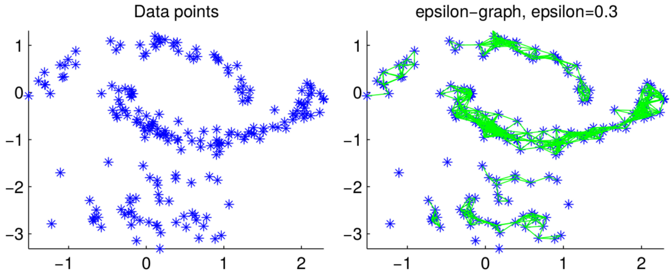
- i and j connected if: $S_{i,j} \leq \epsilon$
- i and j connected: $W_{i,j} = 1$ ($W_{i,j} = 0$ otherwise)



From neighbors to similarity graphs (1/2)

ϵ-neighborhood graph

- i and j connected if: $S_{i,j} \leq \epsilon$
- i and j connected: $W_{i,j} = 1$ ($W_{i,j} = 0$ otherwise)



Remark:

Difficulty: Choosing the radius ϵ

From neighbors to similarity graphs (2/2)

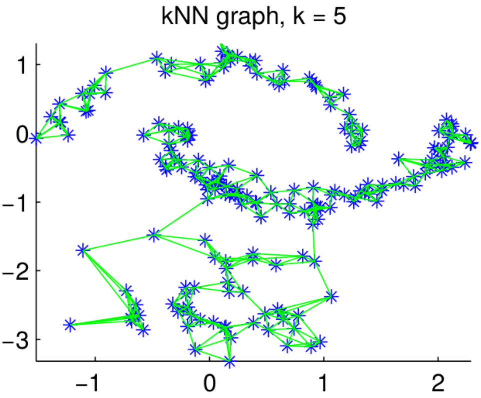
k-Nearest Neighbor graph

- i and j connected if: i among kNN of j or conversely
- i and j connected: $W_{i,j} = 1$ ($W_{i,j} = 0$ otherwise)

From neighbors to similarity graphs (2/2)

k-Nearest Neighbor graph

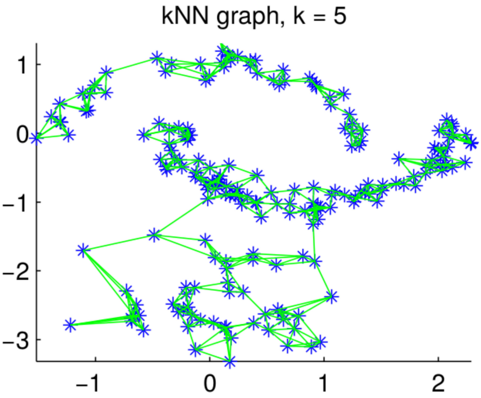
- i and j connected if: i among kNN of j or conversely
- i and j connected: $W_{i,j} = 1$ ($W_{i,j} = 0$ otherwise)



From neighbors to similarity graphs (2/2)

k-Nearest Neighbor graph

- i and j connected if: i among kNN of j or conversely
- i and j connected: $W_{i,j} = 1$ ($W_{i,j} = 0$ otherwise)



Remark:

Difficulty: Choosing the number of neighbors k

Graph structure

Defining a graph G

- **Graph:** $G = (V, E)$
- **Vertices (nodes):** $V = \{v_1, v_2, \ldots, v_n\}$ individuals

Graph structure

Defining a graph G

- **Graph:** $G = (V, E)$

- **Vertices (nodes):** $V = \{v_1, v_2, \ldots, v_n\}$ individuals
  Vertices can be
  - labeled (classes, clusters,...)

Graph structure

Defining a graph G

- **Graph:** $G = (V, E)$

- **Vertices (nodes):** $V = \{v_1, v_2, \ldots, v_n\}$ individuals
  Vertices can be
  - labeled (classes, clusters,...)
  - described by measurements ($v_i \leftrightarrow X_i \in \mathbb{R}^p$)

Graph structure

Defining a graph G

- **Graph:** $G = (V, E)$
- **Vertices (nodes):** $V = \{v_1, v_2, \ldots, v_n\}$ individuals
  Vertices can be
  - labeled (classes, clusters,...)
  - described by measurements ($v_i \leftrightarrow X_i \in \mathbb{R}^p$)
- **Edges:** $E = \{e_{i,j}\}_{1 \leq i,j \leq n}$

Graph structure

Defining a graph G

- **Graph:** $G = (V, E)$
- **Vertices (nodes):** $V = \{v_1, v_2, \ldots, v_n\}$ individuals
  Vertices can be
  - labeled (classes, clusters,...)
  - described by measurements ($v_i \leftrightarrow X_i \in \mathbb{R}^p$)
- **Edges:** $E = \{e_{i,j}\}_{1 \leq i,j \leq n}$
  Edges can be
  - binary valued (connection or no connection): $e_{i,j} \in \{0, 1\}$

Graph structure

Defining a graph G

- **Graph:** $G = (V, E)$

- **Vertices (nodes):** $V = \{v_1, v_2, \ldots, v_n\}$ individuals
  Vertices can be
  - labeled (classes, clusters,. . .)
  - described by measurements ($v_i \leftrightarrow X_i \in \mathbb{R}^p$)

- **Edges:** $E = \{e_{i,j}\}_{1 \leq i,j \leq n}$
  Edges can be
  - binary valued (connection or no connection): $e_{i,j} \in \{0, 1\}$
  - weighted (strength of the link between i and j): $e_{i,j} = W_{i,j} \in \mathbb{R}$

Graph structure

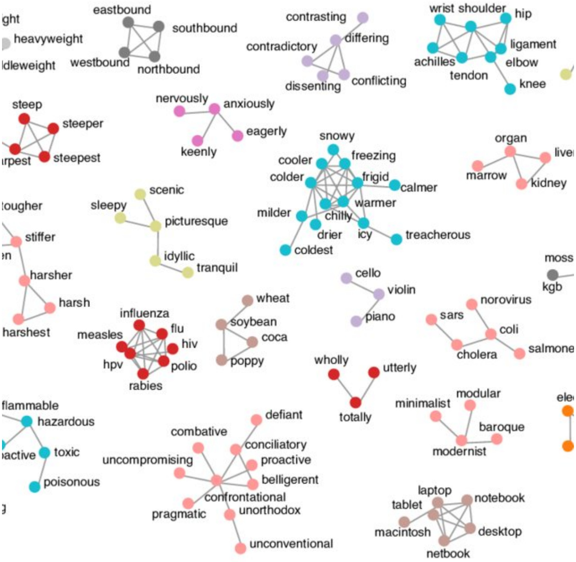
Defining a graph G

- **Graph:** $G = (V, E)$
- **Vertices (nodes):** $V = \{v_1, v_2, \ldots, v_n\}$ individuals
  Vertices can be
  - labeled (classes, clusters,...)
  - described by measurements ($v_i \leftrightarrow X_i \in \mathbb{R}^p$)
- **Edges:** $E = \{e_{i,j}\}_{1 \leq i,j \leq n}$
  Edges can be
  - binary valued (connection or no connection): $e_{i,j} \in \{0, 1\}$
  - weighted (strength of the link between i and j): $e_{i,j} = W_{i,j} \in \mathbb{R}$

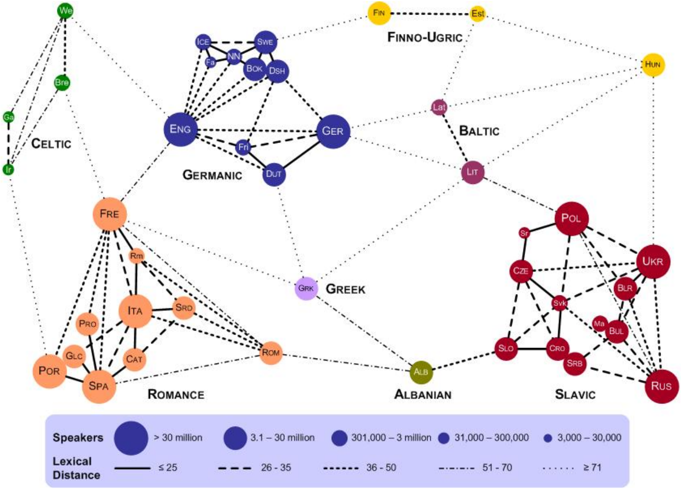
Remark:

No loop assumption means $e_{i,i} = 0$ for all i

Graph with colored vertices (nodes) (1/2)



Graph with colored vertices (nodes) (2/2)



Connected components

Adjacency, Degree, and Laplacian matrix

- $W = (W_{i,j})_{1 \leq i,j \leq d}$: Adjacency matrix

Connected components

Adjacency, Degree, and Laplacian matrix

- $W = (W_{i,j})_{1 \leq i,j \leq d}$: Adjacency matrix
- $D = diag(D_1, \ldots, D_d)$: degree matrix ($D_i = \sum_{j=1}^{d} W_{i,j}$)

Connected components

Adjacency, Degree, and Laplacian matrix

- $W = (W_{i,j})_{1 \leq i,j \leq d}$: Adjacency matrix
- $D = diag(D_1, \ldots, D_d)$: degree matrix ($D_i = \sum_{j=1}^{d} W_{i,j}$)

Connected components

- **Connected component:** largest path containing any pair i, j of nodes

Connected components

Adjacency, Degree, and Laplacian matrix

- $W = (W_{i,j})_{1\leq i,j\leq d}$: Adjacency matrix
- $D = diag(D_1, \dots, D_d)$: degree matrix ($D_i = \sum_{j=1}^{d} W_{i,j}$)

Connected components

- **Connected component:** largest path containing any pair i, j of nodes
- "Connected components" are groups of similar vertices

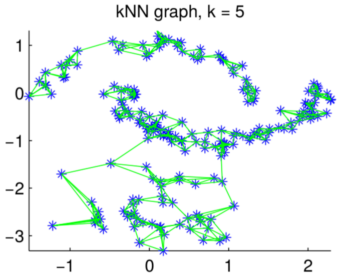
Connected components

Adjacency, Degree, and Laplacian matrix

- $W = (W_{i,j})_{1 \leq i,j \leq d}$: Adjacency matrix
- $D = diag(D_1, \ldots, D_d)$: degree matrix ($D_i = \sum_{j=1}^{d} W_{i,j}$)

Connected components

- **Connected component:** largest path containing any pair i, j of nodes
- "Connected components" are groups of similar vertices



Laplacian matrix

Adjacency, Degree, and Laplacian matrix

- $W = (W_{i,j})_{1 \leq i,j \leq n}$: Adjacency matrix
- $D = diag(D_1, \ldots, D_n)$: degree matrix ($D_i = \sum_{j=1}^{n} W_{i,j}$)

Definition (Laplacian matrix)

- Unnormalized Laplacian:

$$L = D - W$$

- Normalized Laplacian:

$$L_{sym} = (I - D^{-1/2} W D^{-1/2}) \qquad (= D^{-1/2} L D^{-1/2})$$

Idea

- Laplacian matrices yield "Connected components"
- "Connected components" are groups of similar vertices

(Connected component: one path between any pair of vertices)

Connected components

Theorem

If W is nonnegative, then

- $u^\top Lu = \sum_{1\leq i,j\leq n} W_{i,j}(u_i - u_j)^2$, *for all* $u \in \mathbb{R}^d$
- L*: psd matrix*
- *The smallest eigenvalue of L is 0*
- $dim(Null(L)) = k$*: number of connected components*
- *Connect. Comp.:* $A_1, \ldots, A_k$
  $\rightarrow \mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$*: eigenvectors of* $Null(L)$

Proof.

$$u^\top Lu = u^\top Du - u^\top Wu = \frac{1}{2}\left(\sum_{i=1}^{n} u_i^2 D_i - 2\sum_{i,j=1}^{n} u_i u_j W_{i,j} + \sum_{j=1}^{n} u_j^2 D_j\right)$$

$$= \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{i,j}\left[\, u_i^2 - 2u_i u_j + u_j^2 \,\right]\right)$$

Connected components: SVD of L

Strategy

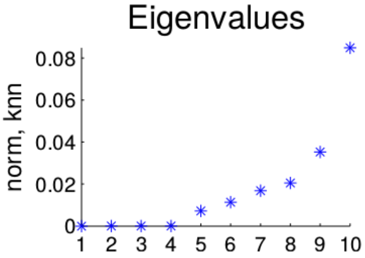
For identifying groups of similar vertices

- Compute the SVD of L
- Find the dimension k of $Null(L)$
- Find the k eigenvetors of $Null(L)$: $u_1, \ldots, u_k$
- For each u_i: non null coordinates $\rightarrow A_i$

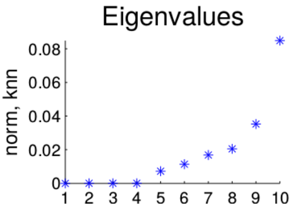
Spectral clustering algorithm

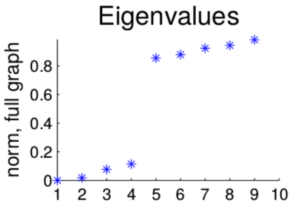
Algorithm: Spectral clustering (unnormalized)

- SVD: $L = U \Lambda U^\top$ $\quad (0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d)$
- Choose k: number of connected components
- $U^{(k)} = [\, u_1, \ldots, u_k \,]$: k "eigenvectors of 0"
- Define $\widetilde{\mathrm{X}} = U^{(k)}$: $d \times k$
- Use k-means for clustering the d rows of $\widetilde{\mathrm{X}}$ (variables) into k clusters $C_1, \ldots, C_k$



Spectral Clustering: Comments

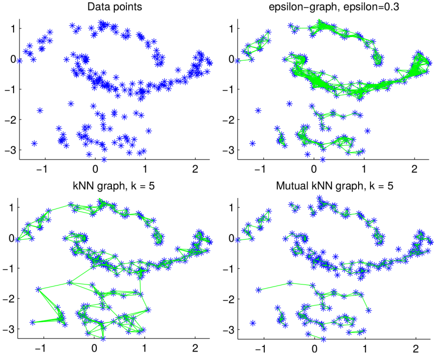




Important remarks

- The choice of k is crucial (number of connected components)
- Not always an easy choice! (see above pictures)
- Eigenvectors can be "noisy" as well (identifiable $Null(L)$)
- Clusters do not necessarily coincide with connected components

Influential parameters: Similarity graphs



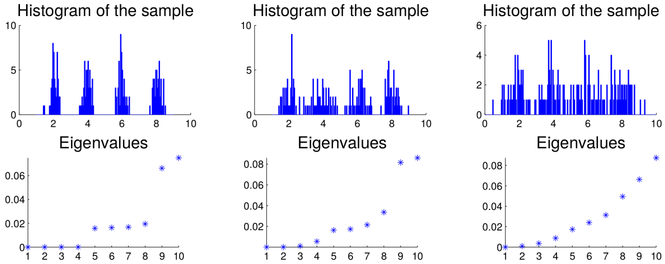
- Similarity is domain dependent: Should be meaningful in the domain of application
- Similarity graph captures non-linear relationships, but depends on parameters (ϵ, kNN,...) (not too small!)

Influential parameters: Number of connected components

Spectral gap heuristic

- Stop at the first largest "Spectral Gap"

$$\widehat{k} = \min\left\{k \mid |\lambda_{k+2} - \lambda_{k+1}| < |\lambda_{k+1} - \lambda_k|\right\}$$



- Not always clear: Depends on the signal-to-noise ratio