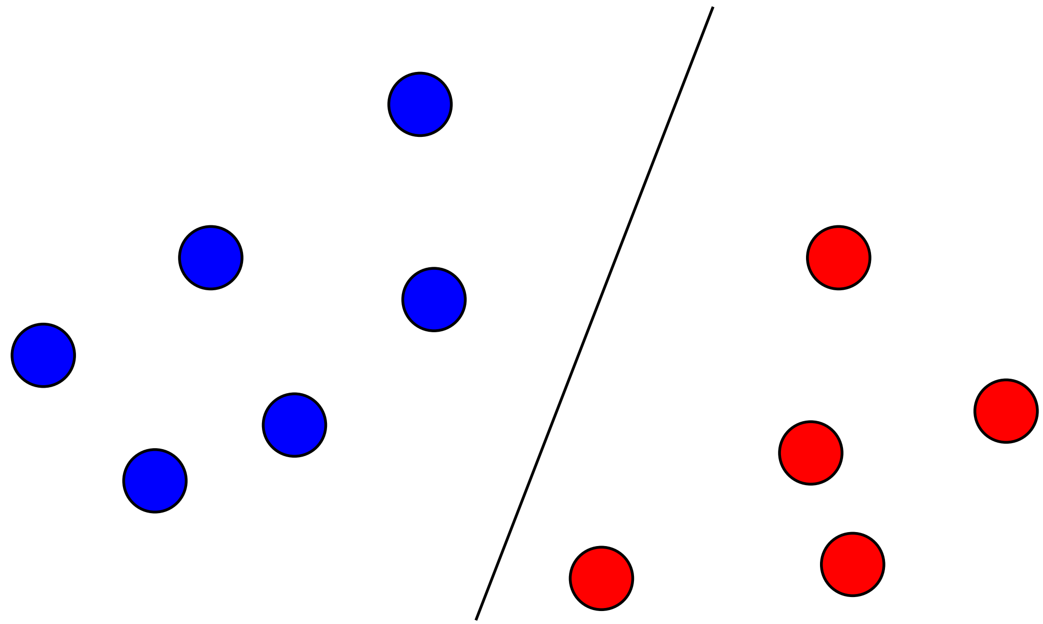
Foundation of Intelligent Systems, Part I

SVM's & Kernel Methods

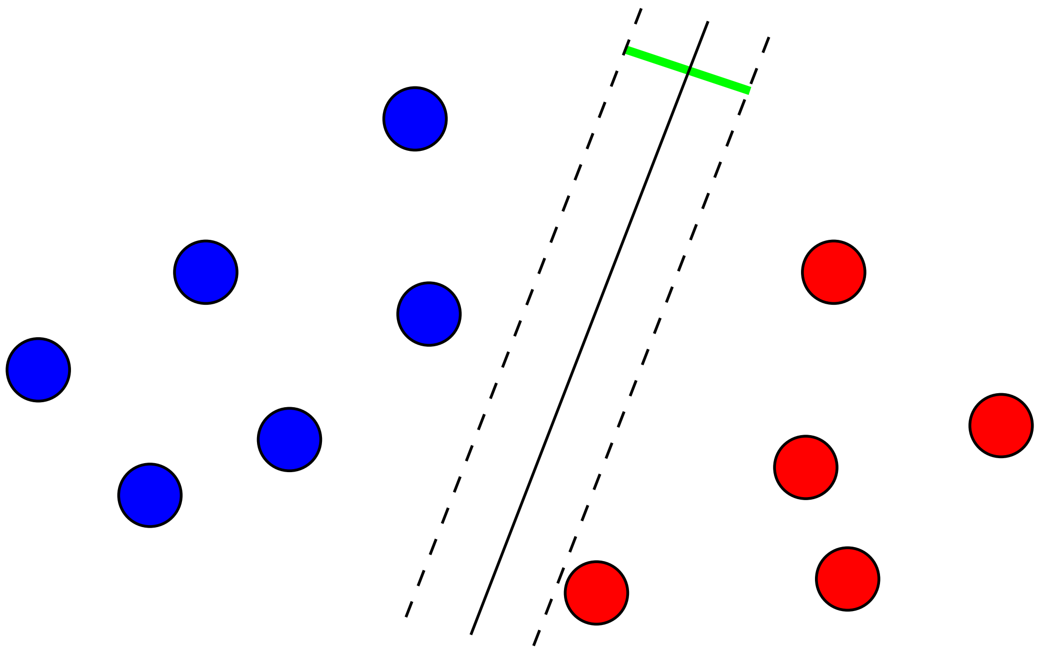
mcuturi@i.kyoto-u.ac.jp

Support Vector Machines
The linearly-separable case

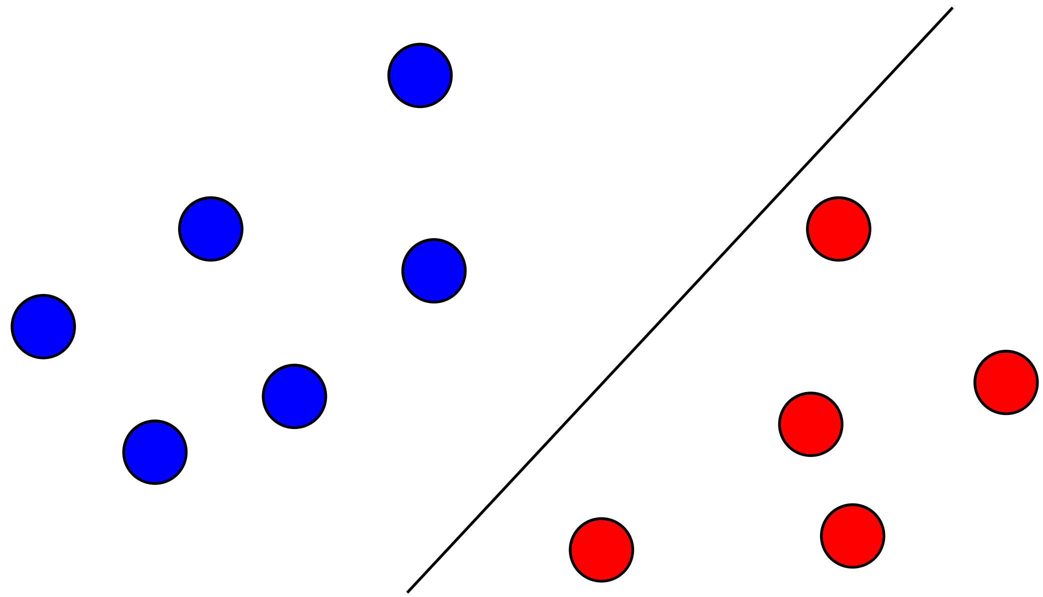
A criterion to select a linear classifier: the margin ?



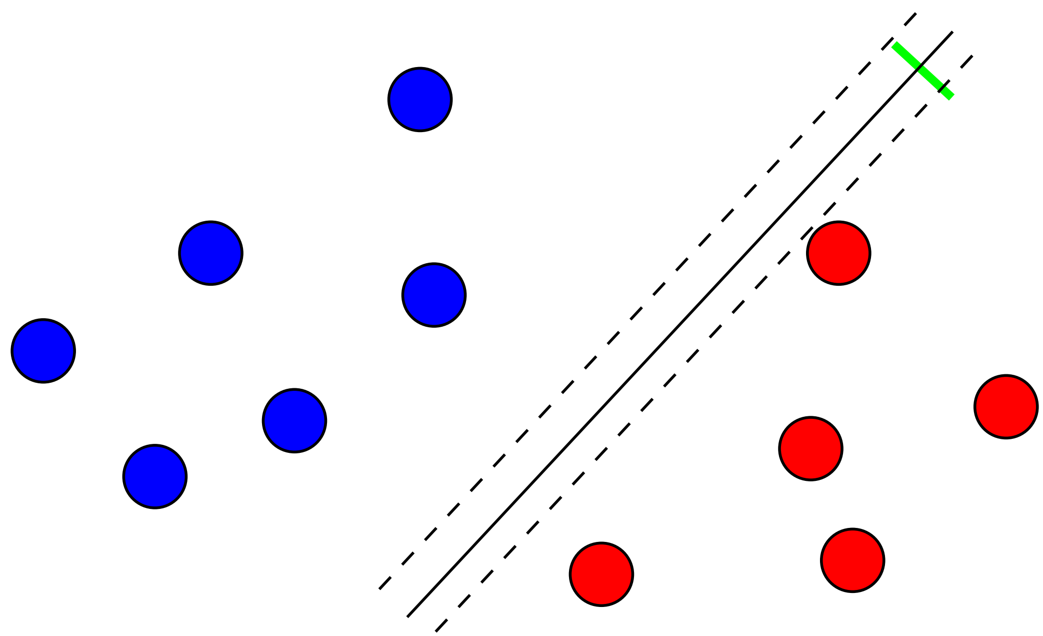
A criterion to select a linear classifier: the margin ?



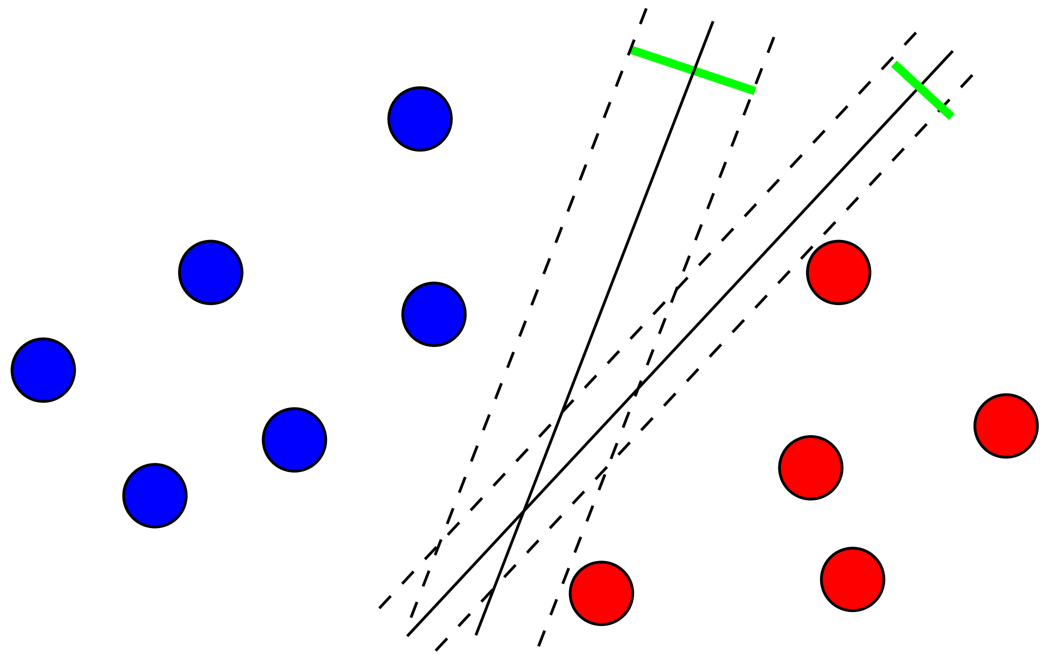
A criterion to select a linear classifier: the margin ?



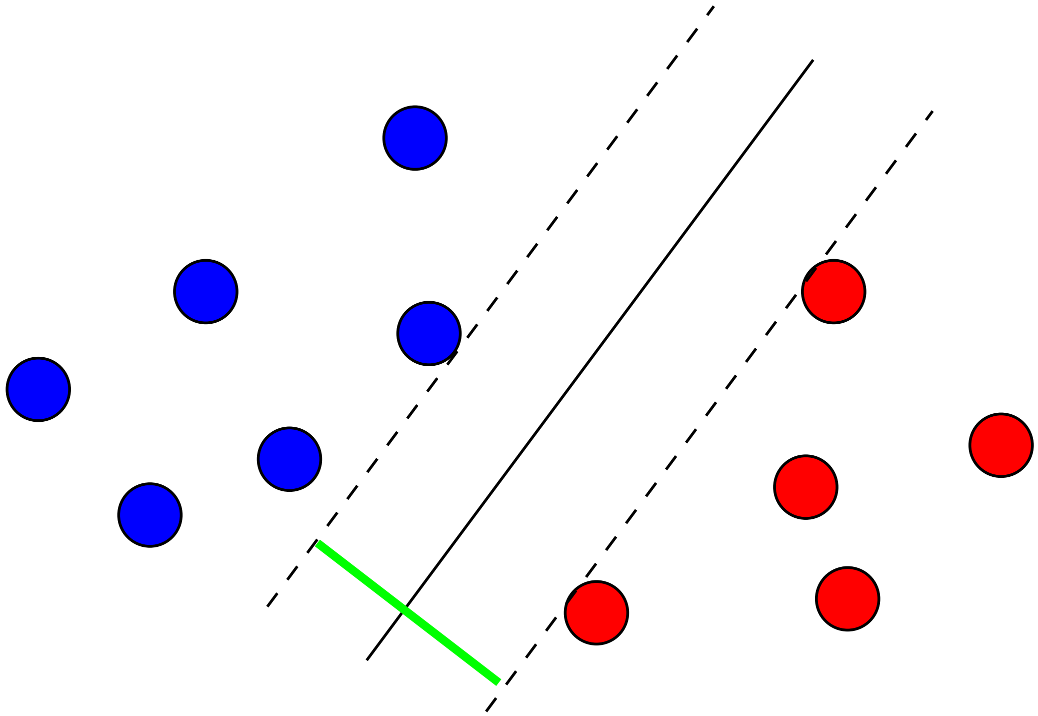
A criterion to select a linear classifier: the margin ?



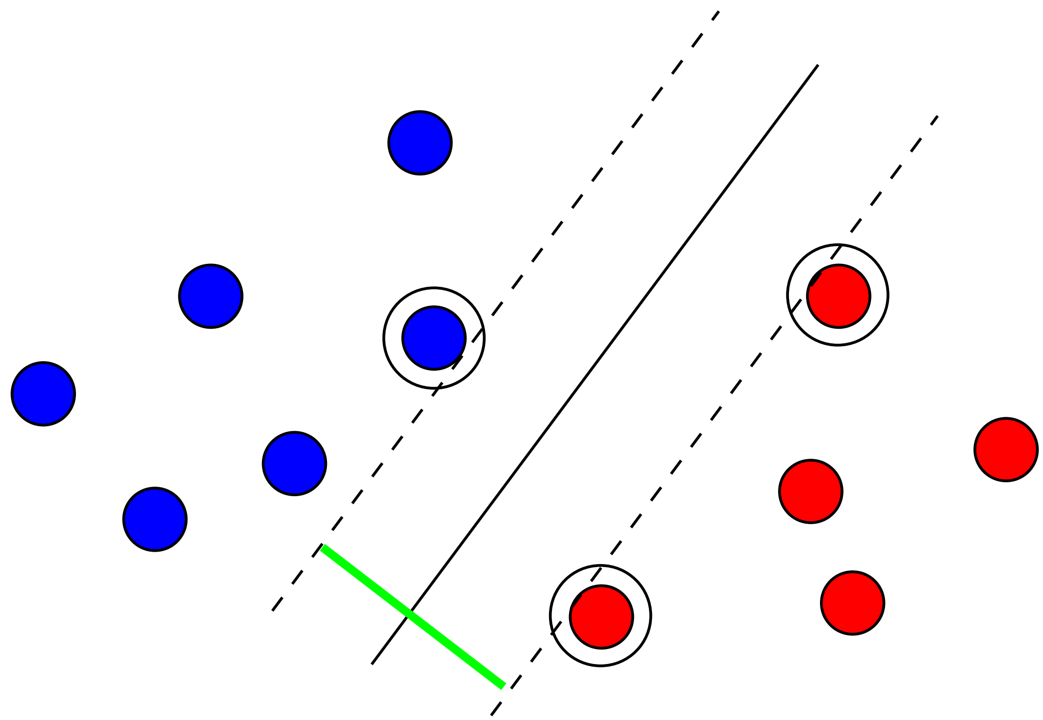
A criterion to select a linear classifier: the margin ?



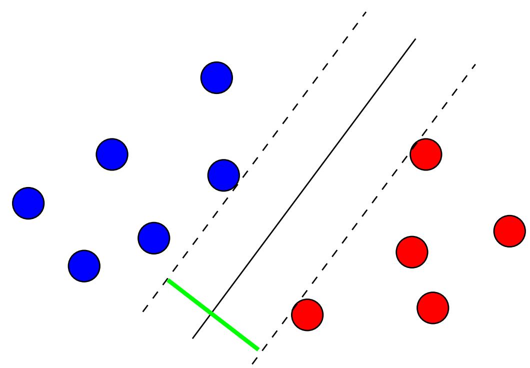
Largest Margin Linear Classifier ?



Support Vectors with Large Margin



Finding the optimal hyperplane



- Finding the optimal hyperplane is equivalent to finding $(\mathbf{w}, b)$ which minimize:

$$\|\mathbf{w}\|^2$$

under the constraints:

$$\forall i = 1, \ldots, n, \qquad \mathbf{y}_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) - 1 \geq 0.$$

This is a classical quadratic program on $\mathbb{R}^{d+1}$
linear constraints - **quadratic objective**

Lagrangian

- In order to minimize:

$$\frac{1}{2}||\mathbf{w}||^2$$

under the constraints:

$$\forall i = 1, \ldots, n, \qquad y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) - 1 \geq 0.$$

- introduce **one dual variable α_i for each constraint**,
- one constraint for **each training point**.
- the **Lagrangian** is, for $\alpha \succeq 0$ (that is for each $\alpha_i \geq 0$)

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) - 1\right).$$

The Lagrange dual function

$$g(\alpha) = \inf_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) - 1 \right) \right\}$$

the saddle point conditions give us that at the minimum in $\mathbf{w}$ and b

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{y}_i \mathbf{x}_i, \quad (\text{ derivating w.r.t } \mathbf{w}) \quad (*)$$

$$0 = \sum_{i=1}^{n} \alpha_i \mathbf{y}_i, \quad (\text{derivating w.r.t } b) \qquad (**)$$

substituting $(*)$ in g, and using $(**)$ as a constraint, get the dual function $g(\alpha)$.

- To solve the dual problem, **maximize** g w.r.t. α.
- **Strong duality holds** : primal and dual problems have the **same optimum**.
- KKT gives us $\alpha_i(\mathbf{y}_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) - 1) = 0$,
  ...*hence*, either $\boldsymbol{\alpha_i = 0}$ or $\boldsymbol{\mathbf{y}_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) = 1}$.
- $\alpha_i \neq 0$ **only** for points on the support hyperplanes $\{(\mathbf{x}, \mathbf{y}) | \, \mathbf{y}_i(\mathbf{w}^T \mathbf{x}_i + b) = 1\}$.

Dual optimum

The dual problem is thus

$$\begin{array}{ll} \text{maximize} & g(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{such that} & \alpha \succeq 0, \sum_{i=1}^n \alpha_i \mathbf{y}_i = 0. \end{array}$$

This is a **quadratic program** in $\mathbb{R}^n$, with *box constraints*.
α^* can be computed using optimization software
($e.g.$ built-in `matlab` function)

Recovering the optimal hyperplane

- With α^*, we recover $(\mathbf{w}^T, b^*)$ corresponding to the **optimal hyperplane**.
- $\mathbf{w}^T$ is given by $\mathbf{w}^T = \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i^T$,
- b^* is given by the conditions on the support vectors $\alpha_i > 0$, $\mathbf{y}_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$,

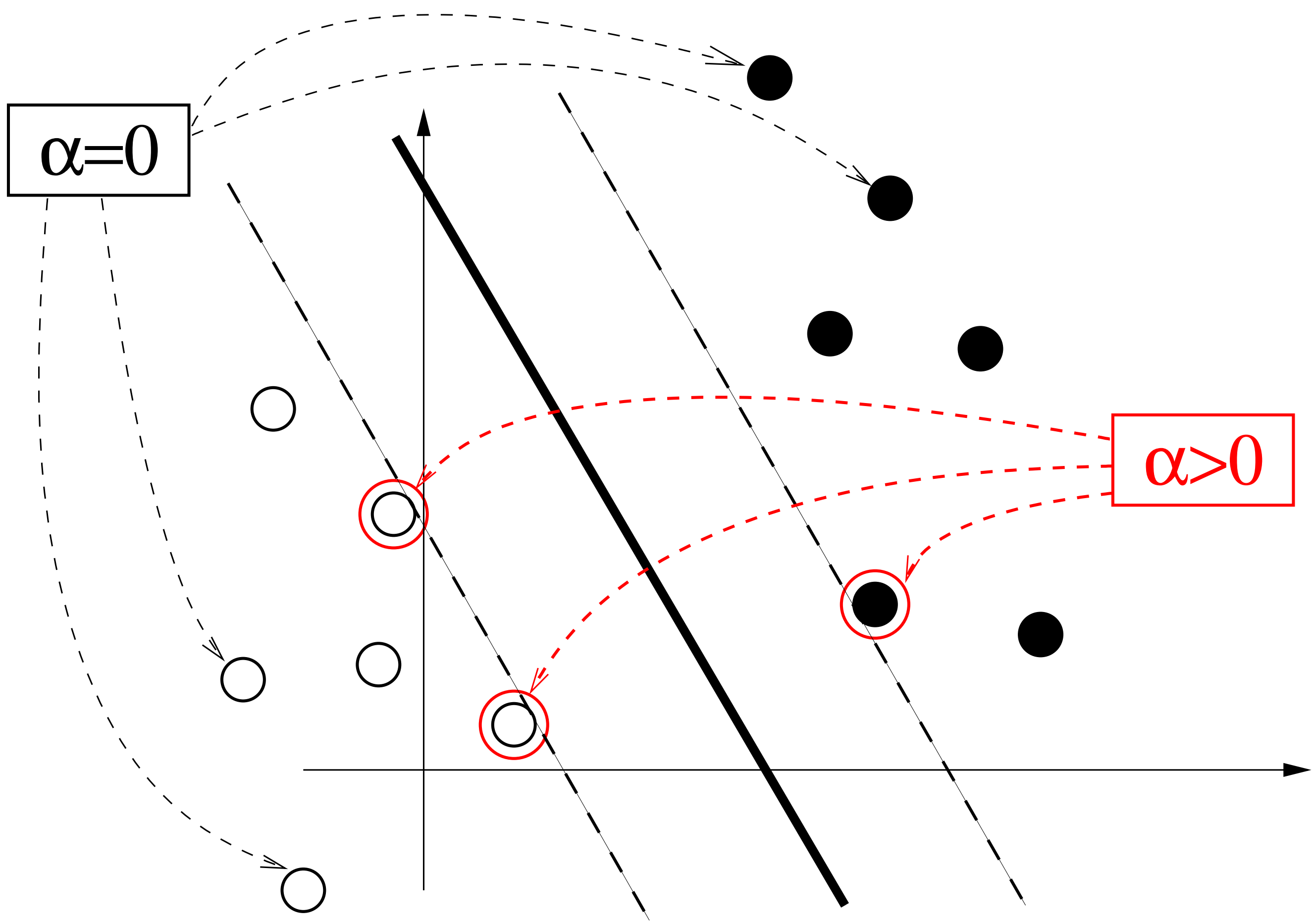
$$b^* = -\frac{1}{2}\left(\min_{\mathbf{y}_i=1,\alpha_i>0}(\mathbf{w}^T\mathbf{x}_i) + \max_{\mathbf{y}_i=-1,\alpha_i>0}(\mathbf{w}^T\mathbf{x}_i)\right)$$

- the **decision function** is therefore:

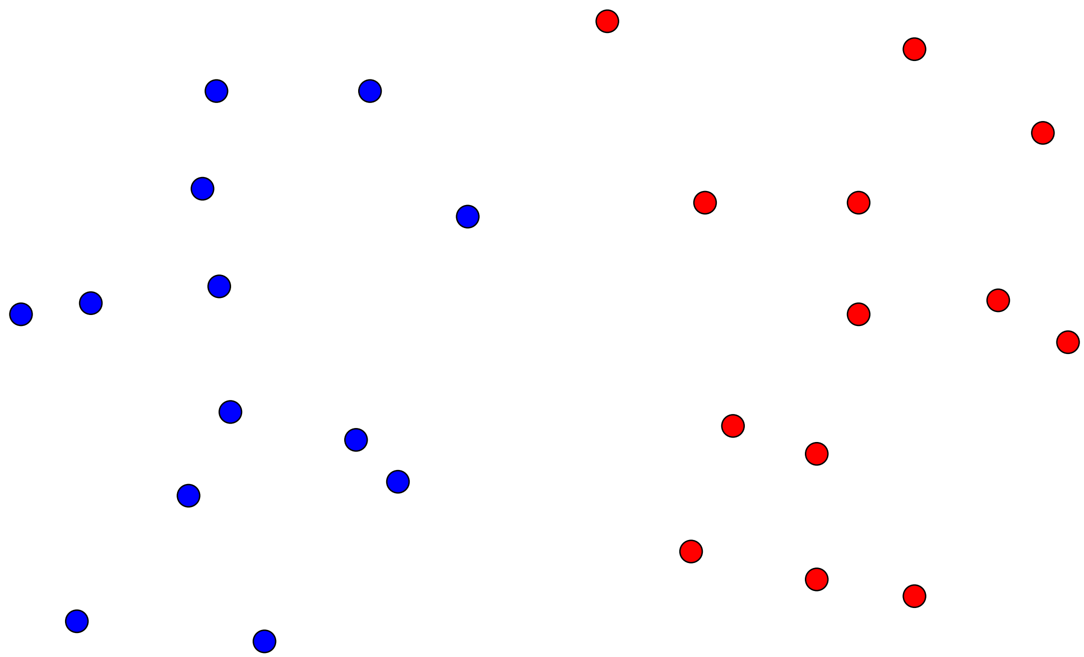
$$\begin{aligned} f^*(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} + b^* \\ &= \left(\sum_{i=1}^n y_i\alpha_i\mathbf{x}_i^T\right)\mathbf{x} + b^*. \end{aligned}$$

- Here the **dual** solution gives us directly the **primal** solution.

Interpretation: support vectors

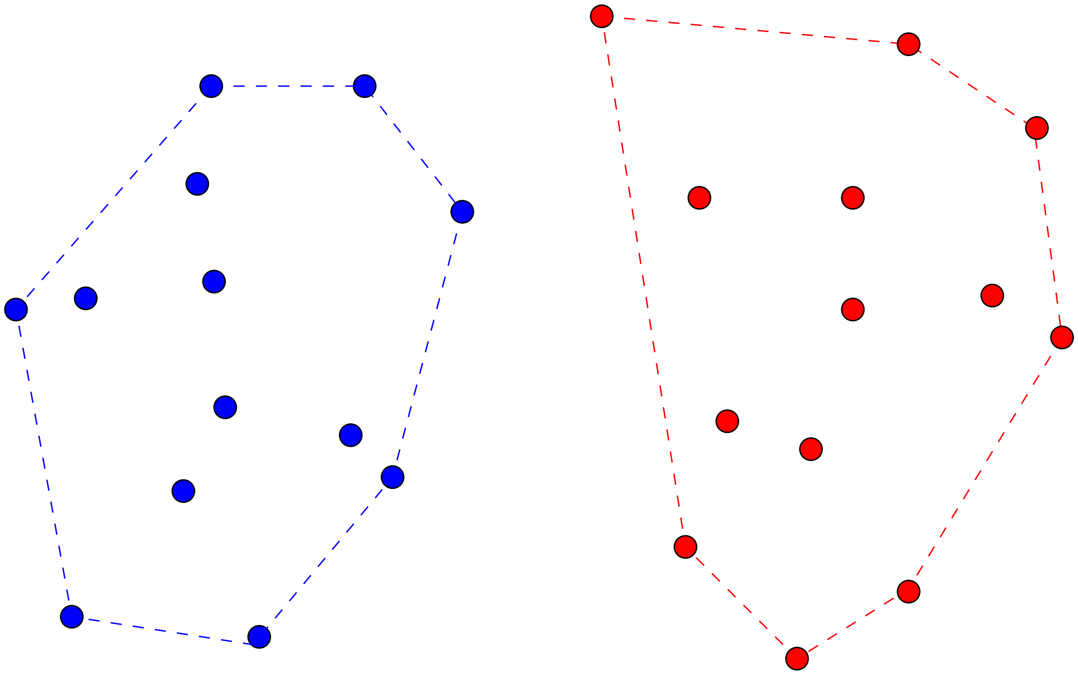


Another interpretation: Convex Hulls



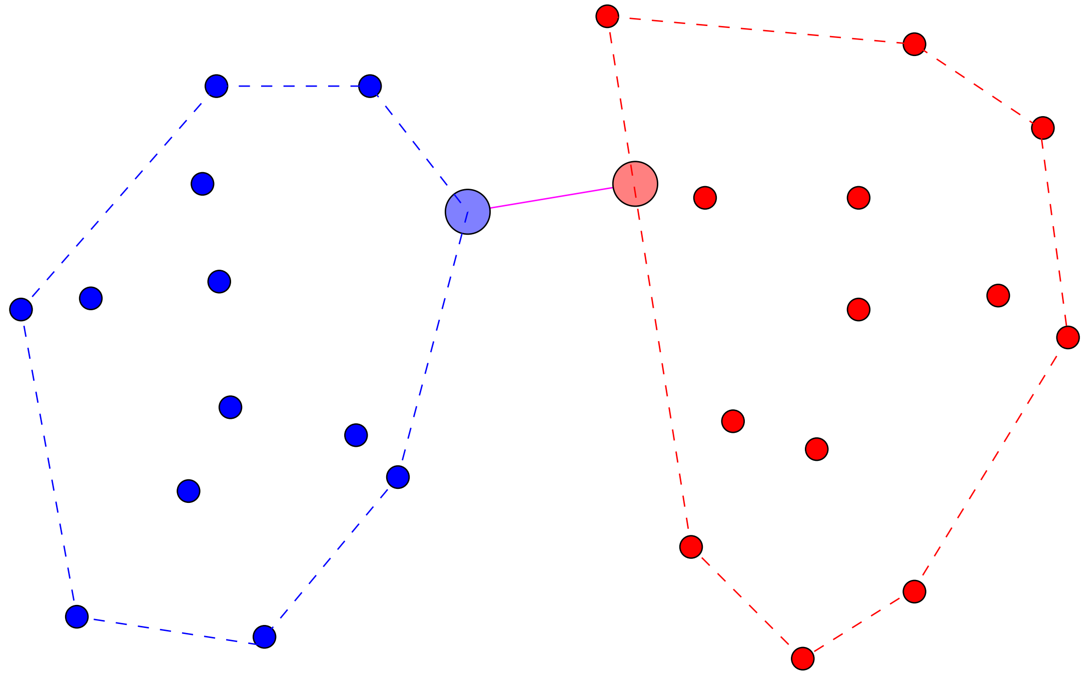
go back to 2 sets of points that are linearly separable

Another interpretation: Convex Hulls



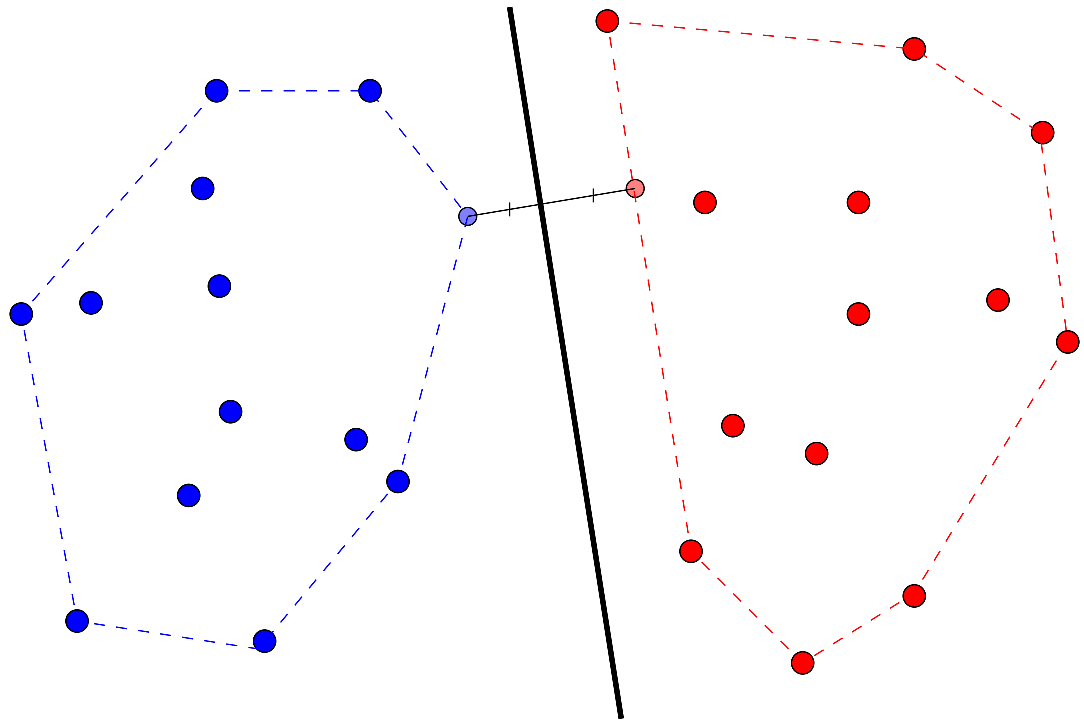
Linearly separable = convex hulls do not intersect

Another interpretation: Convex Hulls



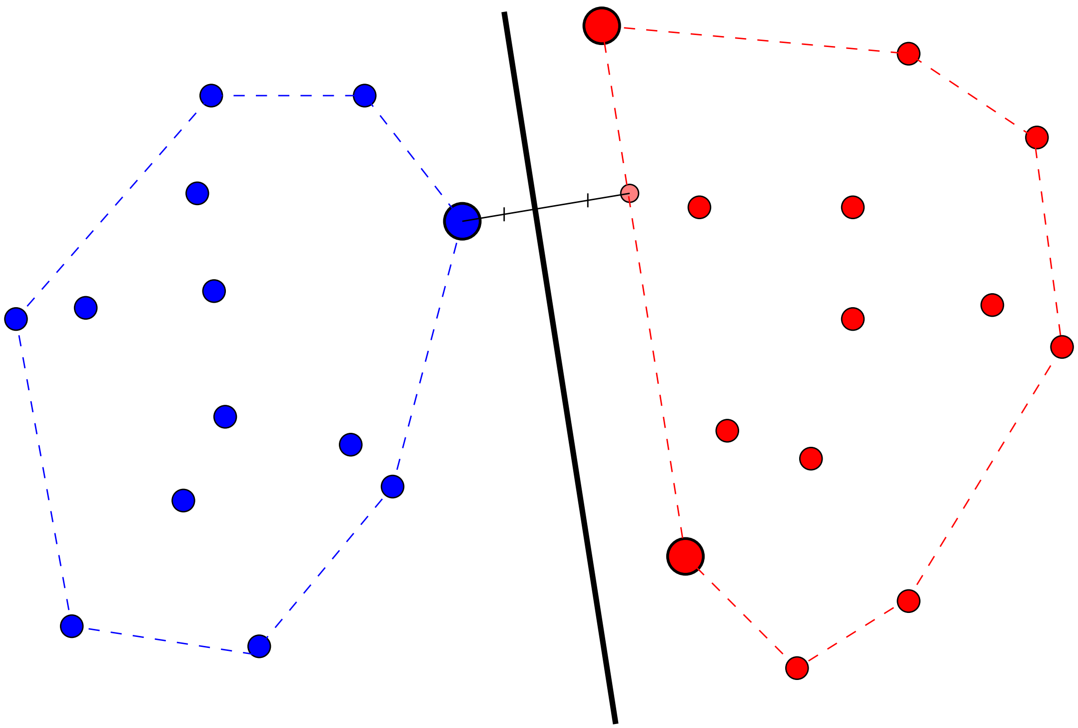
Find two closest points, one in each convex hull

Another interpretation: Convex Hulls



The SVM = bisection of that segment

Another interpretation: Convex Hulls

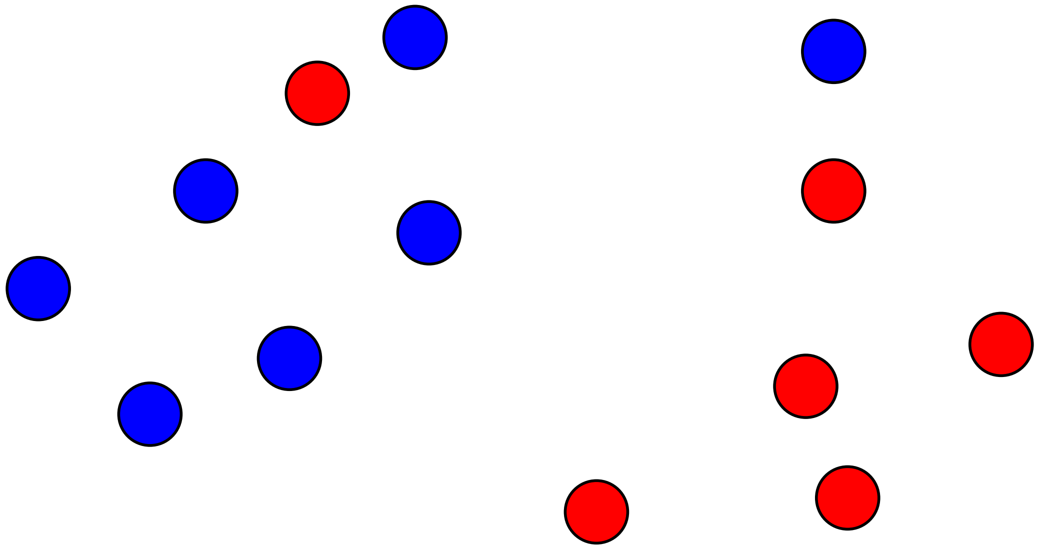


support vectors = extreme points of the faces on which the two points lie

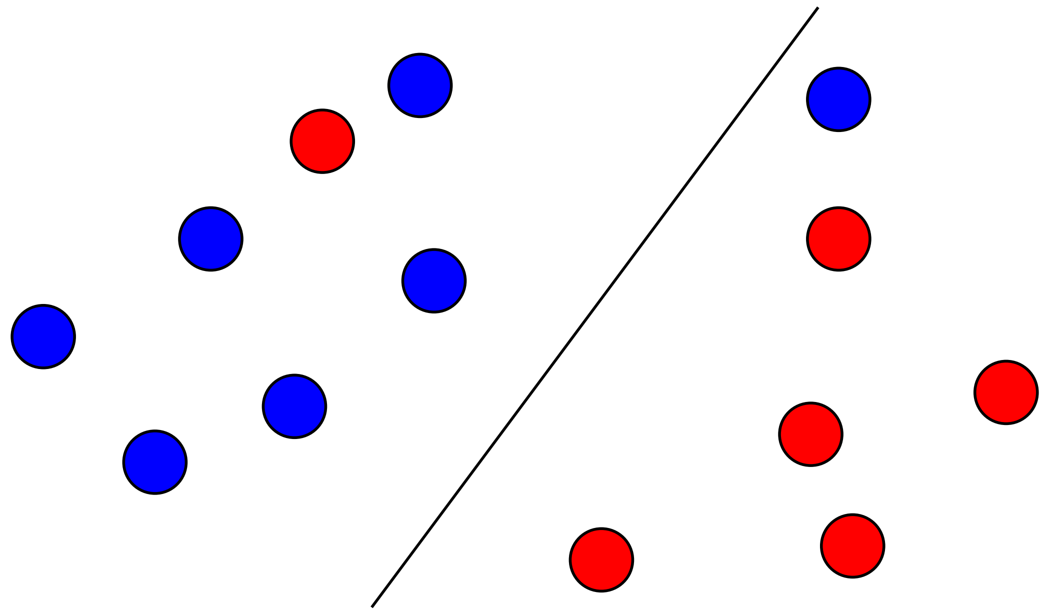
The non-linearly separable case

(when convex hulls intersect)

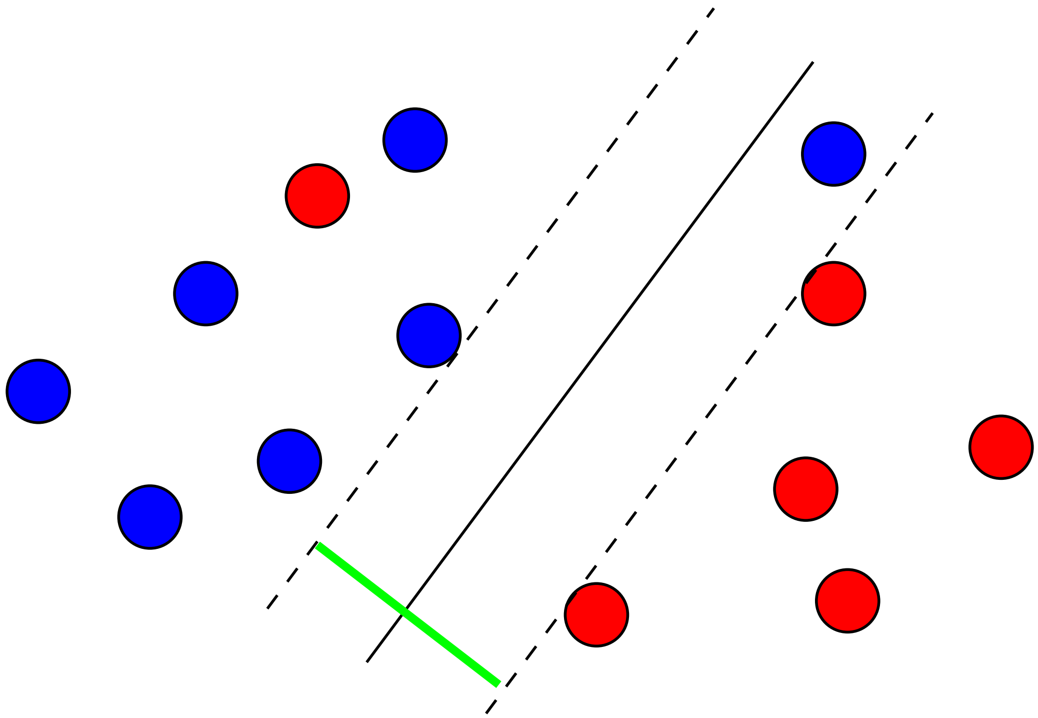
What happens when the data is not linearly separable?



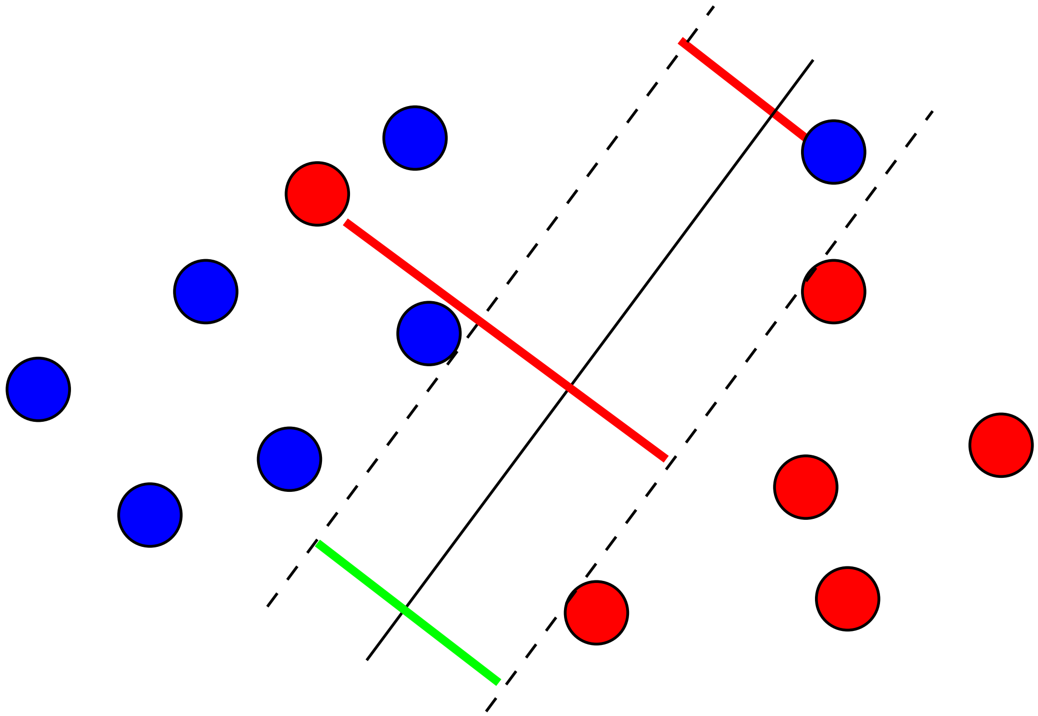
What happens when the data is not linearly separable?



What happens when the data is not linearly separable?



What happens when the data is not linearly separable?



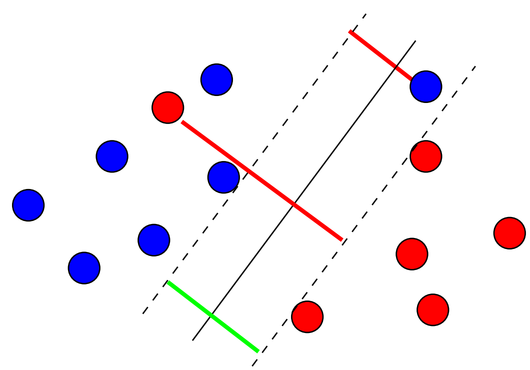
Soft-margin SVM ?

- Find a trade-off between **large margin** and **few errors**.

- Mathematically:

$$\min_f \left\{ \frac{1}{\text{margin}(f)} + C \times \text{errors}(f) \right\}$$

- C is a parameter



Soft-margin SVM formulation ?

- The **margin** of a labeled point $(\mathbf{x}, \mathbf{y})$ is

$$\text{margin}(\mathbf{x}, \mathbf{y}) = \mathbf{y}\left(\mathbf{w}^T\mathbf{x} + b\right)$$

- The **error** is
  - 0 if $\text{margin}(\mathbf{x}, \mathbf{y}) > 1$,
  - $1 - \text{margin}(\mathbf{x}, \mathbf{y})$ otherwise.

- The soft margin SVM solves:

$$\min_{\mathbf{w},b}\{\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\max\{0, 1 - \mathbf{y}_i\left(\mathbf{w}^T\mathbf{x}_i + b\right)\}$$

- $c(u, y) = \max\{0, 1 - yu\}$ is known as the **hinge loss**.

- $c(\mathbf{w}^T\mathbf{x}_i + b, \mathbf{y}_i)$ associates a mistake cost to the decision $\mathbf{w}, b$ for example $\mathbf{x}_i$.

Dual formulation of soft-margin SVM

- The soft margin SVM program

$$\min_{\mathbf{w},b}\{\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\max\{0, 1-\mathbf{y}_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)\}$$

can be rewritten as

$$\begin{array}{ll} \text{minimize} & \|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{such that} & \mathbf{y}_i\left(\mathbf{w}^T\mathbf{x}_i+b\right) \geq 1-\xi_i \end{array}$$

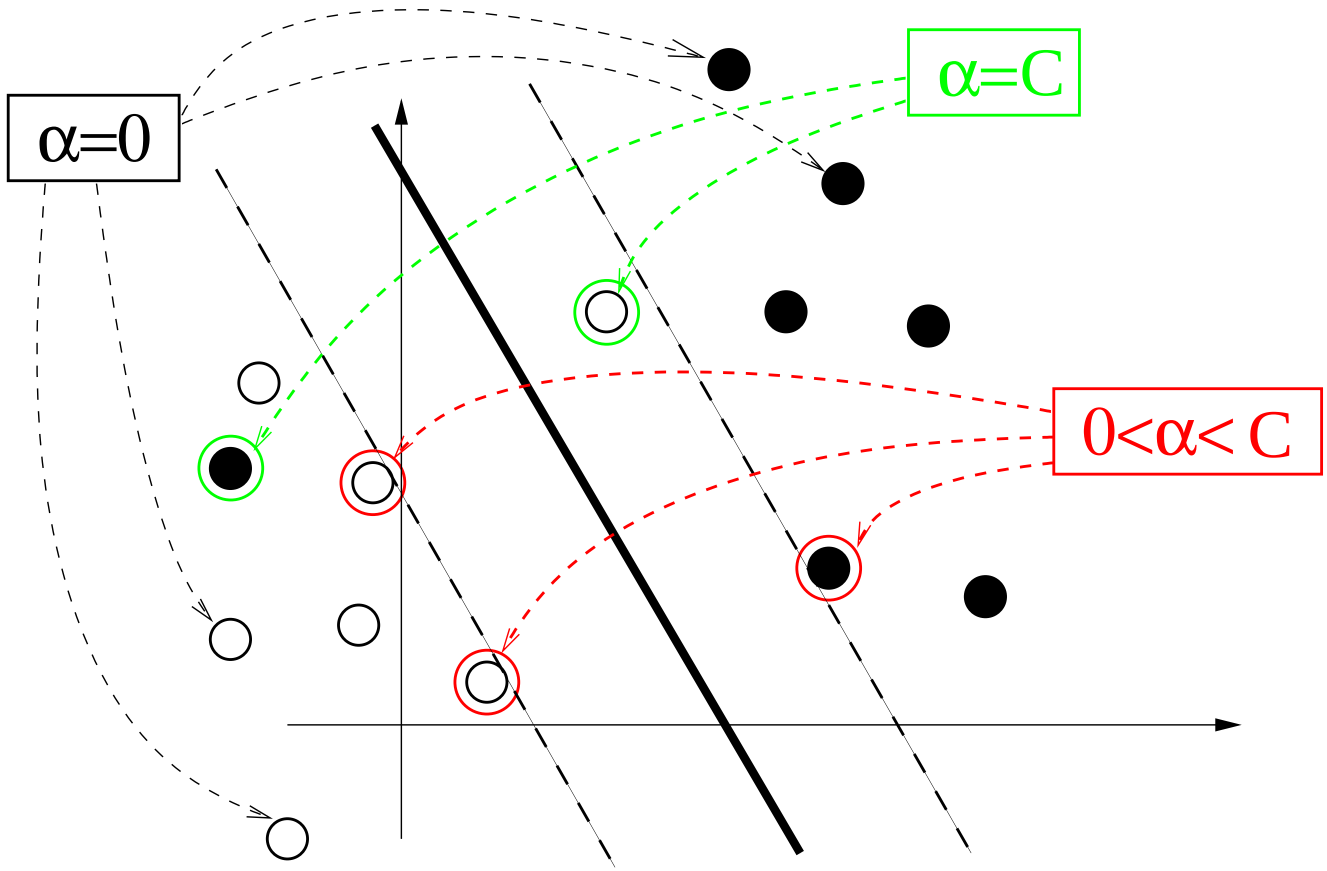
- In that case the dual function

$$g(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j\mathbf{y}_i\mathbf{y}_j\mathbf{x}_i^T\mathbf{x}_j,$$

which is finite under the constraints:

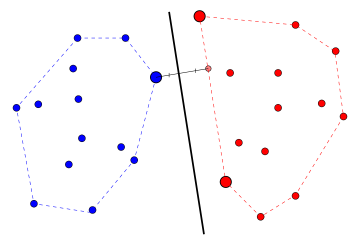
$$\begin{cases} 0 \leq \alpha_i \leq C, & \text{for } i=1,\ldots,n \\ \sum_{i=1}^{n}\alpha_i\mathbf{y}_i = 0. \end{cases}$$

Interpretation: bounded and unbounded support vectors

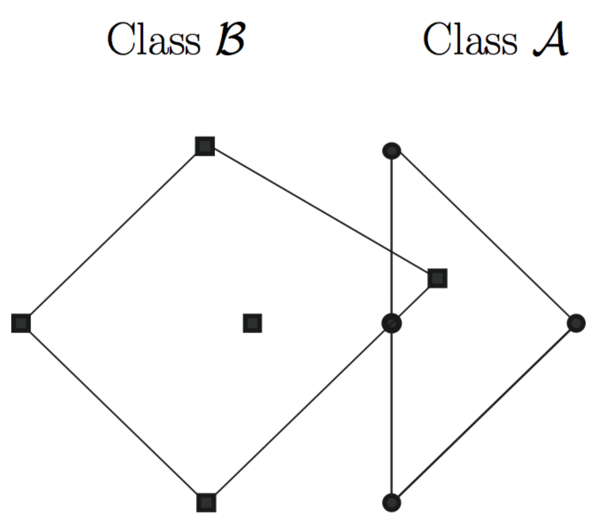


What about the convex hull analogy?

- Remember the separable case



- Here we consider the case where the two sets are not linearly separable, $i.e.$ their convex hulls **intersect.**



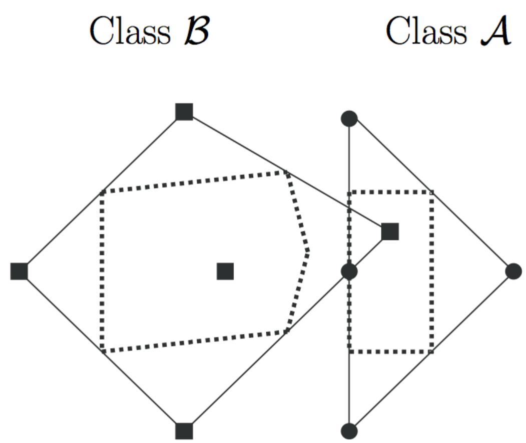
What about the convex hull analogy?

Definition 1. *Given a set of n points $\mathcal{A}$, and $0 \leq C \leq 1$, the set of finite combinations*

$$\sum_{i=1}^{n} \lambda_i \mathbf{x}_i, 1 \leq \lambda_i \leq C, \sum_{i=1}^{n} \lambda_i = 1,$$

is the (C) reduced convex hull of $\mathcal{A}$

- Using $C = 1/2$, the reduced convex hulls of $\mathcal{A}$ and $\mathcal{B}$,



- Soft-SVM with C = closest two points of C-reduced convex hulls.

Kernels

Kernel trick for SVM's

- use a mapping ϕ from $\mathcal{X}$ to a feature space,
- which corresponds to the **kernel** k:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \quad k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$$

- Example: if $\phi(\mathbf{x}) = \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix}$, then

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = (x_1)^2 (x_1')^2 + (x_2)^2 (x_2')^2.$$

Training a SVM in the feature space

Replace each $\mathbf{x}^T\mathbf{x}'$ in the SVM algorithm by $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}')$

- **Reminder**: the dual problem is to maximize

$$g(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{k(\mathbf{x}_i, \mathbf{x}_j)},$$

under the constraints:

$$\begin{cases} 0 \leq \alpha_i \leq C, & \text{for } i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i \mathbf{y}_i = 0. \end{cases}$$

- The **decision function** becomes:

$$\begin{aligned} f(\mathbf{x}) &= \langle \mathbf{w}, \phi(x) \rangle + b^* \\ &= \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{k(\mathbf{x}_i, \mathbf{x})} + b^*. \end{aligned} \tag{1}$$

The Kernel Trick ?

The explicit computation of $\phi(\mathbf{x})$ is not necessary.
The kernel $k(\mathbf{x}, \mathbf{x}')$ is enough.

- the SVM optimization for α works **implicitly** in the feature space.
- the SVM is a kernel algorithm: only need to input $\boldsymbol{K}$ and $\mathbf{y}$:

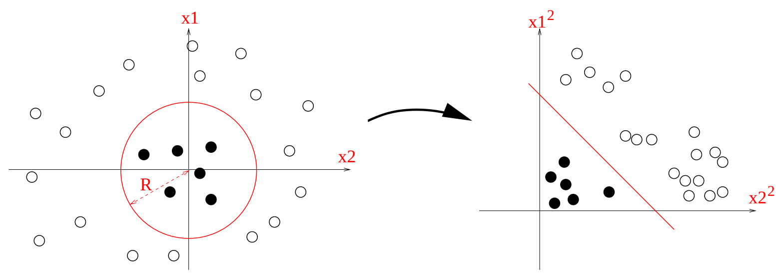
$$\begin{array}{ll} \text{maximize} & g(\alpha) = \alpha^T \mathbf{1} - \frac{1}{2}\alpha^T(\boldsymbol{K} \odot \mathbf{y}\mathbf{y}^T)\alpha \\ \text{such that} & 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i \mathbf{y}_{\boldsymbol{i}} = 0. \end{array}$$

- $\boldsymbol{K}$'s **positive definite** $\Leftrightarrow$ **problem has an unique optimum**
- the decision function is $f(\cdot) = \sum_{i=1}^n \alpha_i \boldsymbol{k}(\mathbf{x}_i, \cdot) + b$.

Kernel example: polynomial kernel

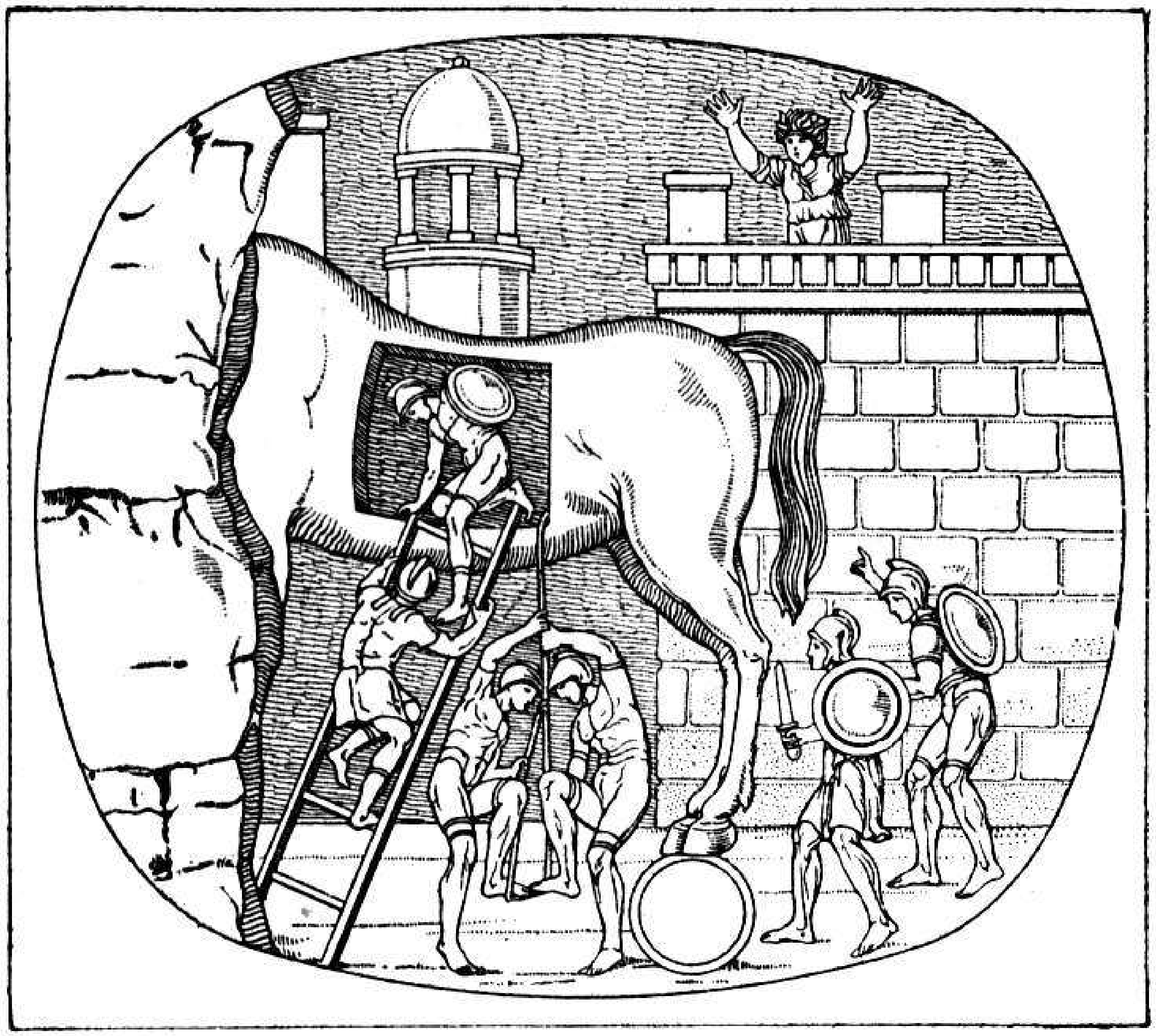
- For $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$, let $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$:

$$
\begin{aligned}
K(\mathbf{x}, \mathbf{x}') &= x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2 \\
&= \{x_1x_1' + x_2x_2'\}^2 \\
&= \{\mathbf{x}^T\mathbf{x}'\}^2 \,.
\end{aligned}
$$



Kernels are Trojan Horses onto Linear Models

- With kernels, complex structures can enter the realm of linear models

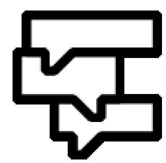
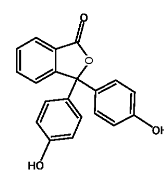


What is a kernel

In the context of these lectures...

- A kernel k is a function

$$
\begin{array}{rccc}
k: & \mathcal{X} \times \mathcal{X} & \longmapsto & \mathbb{R} \\
 & (\mathbf{x}, \mathbf{y}) & \longrightarrow & k(\mathbf{x}, \mathbf{y})
\end{array}
$$

- which compares two objects of a space $\mathcal{X}$, $e.g.$...
  - strings, texts and sequences, 
  - images, audio and video feeds,  
  - graphs, interaction networks and 3D structures 

- whatever actually... time-series of graphs of images? graphs of texts?...

Fundamental properties of a kernel

symmetric

$$k(\mathbf{x},\mathbf{y}) = k(\mathbf{y},\mathbf{x}).$$

positive-(semi)definite

for any *finite* family of points $\mathbf{x}_1, \cdots, \mathbf{x}_n$ of $\mathcal{X}$, the matrix

$$K = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & \cdots & k(\mathbf{x}_1,\mathbf{x}_i) & \cdots & k(\mathbf{x}_1,\mathbf{x}_n) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & \cdots & k(\mathbf{x}_2,\mathbf{x}_i) & \cdots & k(\mathbf{x}_2,\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ k(\mathbf{x}_i,\mathbf{x}_1) & k(\mathbf{x}_i,\mathbf{x}_2) & \cdots & k(\mathbf{x}_i,\mathbf{x}_i) & \cdots & k(\mathbf{x}_2,\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n,\mathbf{x}_1) & k(\mathbf{x}_n,\mathbf{x}_2) & \cdots & k(\mathbf{x}_n,\mathbf{x}_i) & \cdots & k(\mathbf{x}_n,\mathbf{x}_n) \end{bmatrix} \succeq 0$$

is positive semidefinite (has a nonnegative spectrum).

K is often called the **Gram matrix** of $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ using k

What can we do with a kernel?

The setting

- Pretty simple setting: a set of objects $\mathbf{x}_1, \cdots, \mathbf{x}_n$ of $\mathcal{X}$
- **Sometimes** additional information on these objects
  - labels $\mathbf{y}_i \in \{-1, 1\}$ or $\{1, \cdots, \#(\text{classes})\}$,
  - scalar values $\mathbf{y}_i \in \mathbb{R}$,
  - associated object $\mathbf{y}_i \in \mathcal{Y}$

- A kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$.

A few intuitions on the possibilities of kernel methods

Important concepts and perspectives

- The functional perspective: represent **points as functions**.
- **Nonlinearity** : linear combination of kernel evaluations.
- Summary of a sample through its **kernel matrix**.

Represent any point in $\mathcal{X}$ as a function

For every $\mathbf{x}$, the map
$$\mathbf{x} \longrightarrow k(\mathbf{x}, \cdot)$$
associates to $\mathbf{x}$ a function $k(\mathbf{x}, \cdot)$ from $\mathcal{X}$ to $\mathbb{R}$.

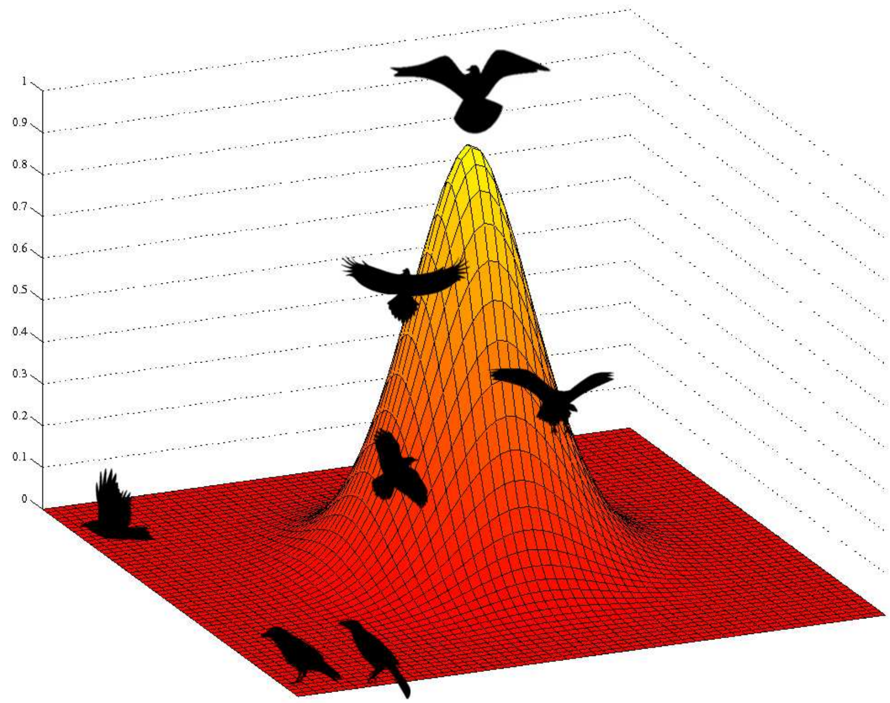
- Suppose we have a kernel k on bird images

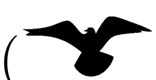
- Suppose for instance

$k\,($  $,$  $) = .32$

Represent any point in $\mathcal{X}$ as a function

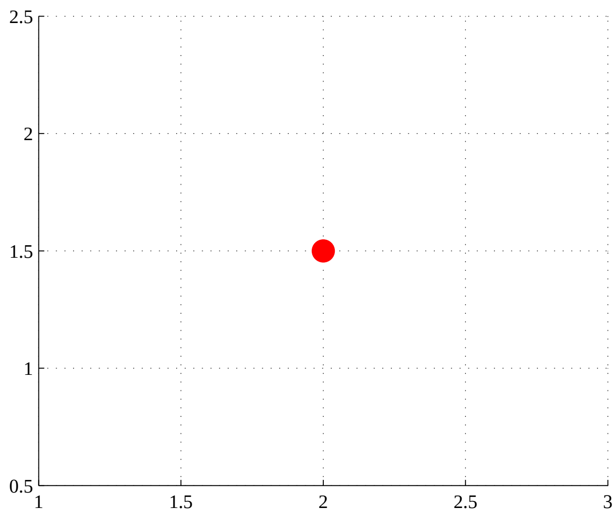
- We examine one image in particular: 
- With kernels, we get a **representation** of that bird as a real-valued function, defined on the space of birds, represented here as $\mathbb{R}^2$ for simplicity.



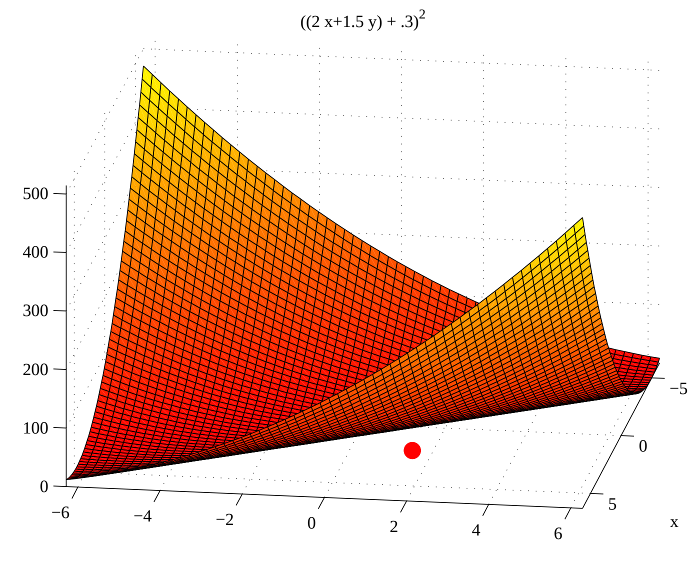
schematic plot of $k($  $, \cdot\ )$.

Represent any point in $\mathcal{X}$ as a function

- If the bird example was confusing...
- $k\left(\begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} x' \\ y' \end{bmatrix}\right) = \left(\begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} x' \\ y' \end{bmatrix} + .3\right)^2$
- From a point in $\mathbb{R}^2$ to a function defined over $\mathbb{R}^2$.



$\rightarrow$



- We assume implicitly that the **functional representation** will be more useful than the **original representation**.

Decision functions as linear combination of kernel evaluations

- Linear decisions functions are a major tool in statistics, that is functions

$$f(\mathbf{x}) = \beta^T \mathbf{x} + \beta_0.$$

- Implicitly, a point $\mathbf{x}$ is processed depending on its characteristics x_i,

$$f(\mathbf{x}) = \sum_{i=1}^{d} \boldsymbol{\beta_i} x_i + \boldsymbol{\beta_0}.$$

  the free parameters are scalars $\boldsymbol{\beta_0, \beta_1, \cdots, \beta_d}$.

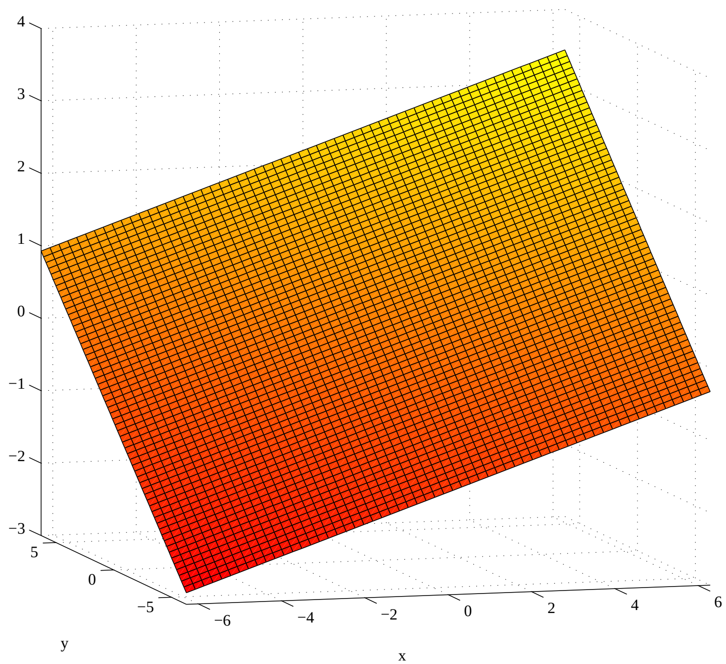
- Kernel methods yield candidate decision functions

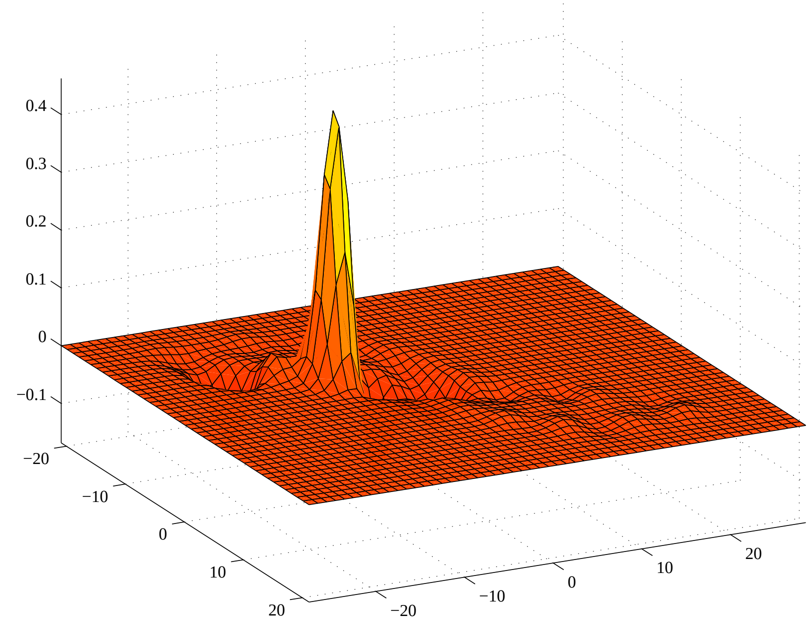
$$f(\mathbf{x}) = \sum_{j=1}^{\boldsymbol{n}} \boldsymbol{\alpha_j} k(\mathbf{x}_j, \mathbf{x}) + \boldsymbol{\alpha_0}.$$

  the free parameters are scalars $\boldsymbol{\alpha_0, \alpha_1, \cdots, \alpha_n}$.

Decision functions as linear combination of kernel evaluations

- linear decision surface / linear expansion of **kernel surfaces** (here $k_G(\mathbf{x}_i, \cdot)$)





- Kernel methods are considered **non-linear** tools.
- Yet not completely “nonlinear” $\rightarrow$ only one-layer of nonlinearity.

kernel methods use the data as a functional base to define decision functions

Decision functions as linear combination of kernel evaluations

database $\{\mathbf{x}_i, i = 1, \ldots, N\}$

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \; k\left(\mathbf{x}_i, \mathbf{x}\right)$$

kernel definition

weights α estimated
with a kernel machine

- f is any predictive function of interest of a new point **x**.
- Weights α are **optimized** with a kernel machine ($e.g.$ support vector machine)

intuitively, kernel methods provide decisions based on how *similar* a point x is to each instance of the training set

The Gram matrix perspective

- Imagine a little task: you have read 100 novels so far.

- You would like to know whether you will enjoy reading a **new** novel.
- A few options:
  - read the book...
  - have friends read it for you, read reviews.
  - try to guess, based on the novels you read, if you will like it

The Gram matrix perspective

Two distinct approaches

- Define what **features** can characterize a book.
  - Map each book in the library onto vectors

$$\longrightarrow \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

typically the x_i's can describe...

▷ # pages, language, year 1st published, country,
▷ coordinates of the main action, keyword counts,
▷ author's prizes, popularity, booksellers ranking

- Challenge: find a decision function using 100 ratings and features.

The Gram matrix perspective

- Define what makes **two novels similar**,
  - Define a kernel k which quantifies novel similarities.
  - Map the library onto a Gram matrix

$$\longrightarrow \quad K = \begin{bmatrix} k(b_1, b_1) & k(b_1, b_2) & \cdots & k(b_1, b_{100}) \\ k(b_2, b_1) & k(b_2, b_2) & \cdots & k(b_2, b_{100}) \\ \vdots & \vdots & \ddots & \vdots \\ k(b_n, b_1) & k(b_n, b_2) & \cdots & k(b_{100}, b_{100}) \end{bmatrix}$$

- Challenge: find a decision function that takes this 100×100 matrix as an input.

The Gram matrix perspective

Given a new novel,

- with the **features approach**, the prediction can be rephrased as **what are the features of this new book**? what **features** have I found in the past that were good indicators of my taste?

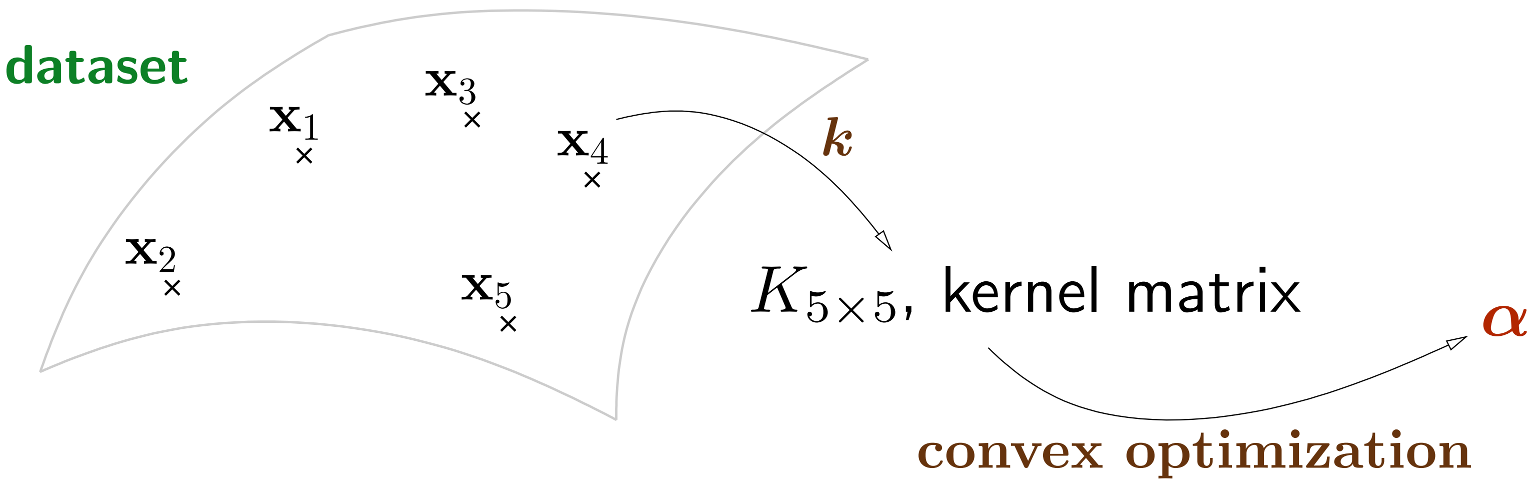
- with the **kernel approach**, the prediction is rephrased as **which novels this book is similar or dissimilar to?** what **pool of books** did I find the most influentials to define my tastes accurately?

kernel methods **only use kernel similarities**, do not consider features.

Features can help define similarities, but **never considered elsewhere**.

The Gram matrix perspective

in kernel methods, clear separation between the kernel...



and **Convex optimization** (thanks to psdness of K, more later) to output the α's.

Mathematical Considerations on Kernels

different definitions and properties of the same mathematical object

space of functions

- In the next slides we focus on

reproducing kernel **Hilbert spaces** (RKHS)

- This term is ubiquitous in the kernel methods literature.
- "Old" mathematics [Mer09], [Aro50]. Survey in [BTA03].

- Reminder: a **Hilbert space** is a
  - vector space, possibly infinite dimensional,
  - equipped with a dot-product, $i.e.$
    - ▷ a bilinear symmetric application
    - ▷ which satisfies $\langle x, x\rangle \geq 0$, equal to 0 only with $x = 0$.
  - complete (all Cauchy sequences **converge** inside the space).
- **reproducing kernel**... a new term.

reproducing kernels

- Let $\mathcal{H}$ be a Hilbert space of real-valued functions on $\mathcal{X}$.

Definition 2 (RKHS)**.** *$\mathcal{H}$ is said to be a reproducing kernel Hilbert space if every linear map of the form $L_{\mathbf{x}} : f \mapsto f(\mathbf{x})$ from $\mathcal{H}$ to $\mathbb{R}$ is continuous for any $\mathbf{x}$ in $\mathcal{X}$.*

Where is the **reproducing kernel** in this definition?

reproducing kernels

- By the **Riesz representation theorem**
  - *Any **continuous** **linear** functional $L(\cdot)$ on $\mathcal{H}$ can be written uniquely $\langle \mathbf{u}, \cdot \rangle_{\mathcal{H}}$*

  we hence have that:

$$\forall \mathbf{x} \in \mathcal{X}, \exists ! \, k_{\mathbf{x}} \in \mathcal{H} \quad | \quad f(\mathbf{x}) = \langle f, \; k_{\mathbf{x}} \rangle_{\mathcal{H}}, \quad \forall f \in \mathcal{H}$$

  $k_{\mathbf{x}}$ is called the point-evaluation functional at the point $\mathbf{x}$.

- Since $\mathcal{H}$ is a space of functions, $k_{\mathbf{x}}$ is itself a function. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined by

$$k(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} k_{\mathbf{x}}(\mathbf{y}).$$

- k is the **reproducing kernel** of $\mathcal{H}$ and it is determined entirely by $\mathcal{H}$ through the Riesz representation theorem which guarantees the **unicity** of $k_{\mathbf{x}}$ for each $\mathbf{x}$.

positive definite kernels

Definition 3 (Real-valued Positive Definite Kernels). *A symmetric function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a positive definite (p.d.) kernel on* $\mathcal{X}$ *if*

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

holds for any $n \in \mathbb{N}, x_1, \ldots, x_n \in \mathcal{X}$ *and* $c_1 \ldots, c_n \in \mathbb{R}$.

With this definition, the set of p.d. kernels $\mathcal{P}(\mathcal{X})$ is a closed, convex pointed cone:

- $\forall \lambda \geq 0, k$ p.d.kernel $\Rightarrow \lambda k$ is p.d.
- $\forall \lambda \geq 0, k_1, k_2$ p.d.kernel, $\lambda k_1 + (1-\lambda) k_2$ p.d. kernel.
- k p.d. kernel, $-k$ p.d. kernel $\Rightarrow k = 0$.
- if $k_n \in \mathcal{P}(\mathcal{X})$ and $\lim_{n\infty} k_n = k$ then $k \in \mathcal{P}(\mathcal{X})$.

kernels: two definitions

- Is there an ambiguity here?

reproducing kernels (functional analysis, topology)

$$\stackrel{?}{\neq}$$

positive definite kernels (positivity and linear algebra)

- luckily, no screw up: the two notions are equivalent.

Moore-Aronszajn (1950) theorem

Theorem 1. *Let $\mathcal{X}$ be any set. An application $\mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a reproducing kernel iff it is a positive definite kernel*

- A first proof was given by Mercer (1909) when $\mathcal{X}$ is compact.
- Hence the *Mercer* kernel term sometimes used.
- In many applications compacity is never really mentioned...
- ... hence *positive definite* or *reproducing* are more accurate terms.
- In the general case the result was proved by Moore & Aronszajn in 1950 (separately).

Moore-Aronszajn (1950) theorem, proof outline

- If k is a r.k., $k(\mathbf{x},\mathbf{y}) = \langle k(\mathbf{x},\cdot), k(\mathbf{y},\cdot)\,\rangle = \langle k(\mathbf{y},\cdot), k(\mathbf{x},\cdot)\,\rangle = k(\mathbf{y},\mathbf{x})$,

$$\sum_{i,j=1}^{n} c_i c_j k(\mathbf{x}_i,\mathbf{x}_j) = \left\| \sum_{i=1}^{n} k(\mathbf{x}_i,\cdot) \right\|_{\mathcal{H}}^2 \geq 0.$$

- if k is a p.d. kernel,
  - Define the vector space $\tilde{\mathcal{H}} = \operatorname{span}\{k(\mathbf{x},\cdot)\}$.
  - Define $\langle\cdot,\cdot\rangle_{\tilde{\mathcal{H}}}$ for $f = \sum_{i=1}^{m} \alpha_i k(\mathbf{x}_i,\cdot)$ and $g = \sum_{j=1}^{n} \beta_j k(\mathbf{y}_j,\cdot)$ as

$$\langle f, g\,\rangle = \sum_{i,j=1}^{m,n} \alpha_i \beta_j k(\mathbf{x}_i,\mathbf{y}_j).$$

  - even if $\{k(\mathbf{x},\cdot)\}_{\mathbf{x}\in\mathcal{X}}$ is not a l.i. family ($i.e.$ no unicity of α or β) we have

$$\langle f, g\,\rangle = \sum_{i=1}^{m} \alpha_i g(\mathbf{x}_i) = \sum_{j=1}^{n} \beta_i f(\mathbf{y}_i).$$

- $\langle \cdot, \cdot \rangle_{\tilde{\mathcal{H}}}$ is **bilinear symmetric** and **p.d.** through the p.d. of k.
- Cauchy-Schwartz is verified thanks to p.d. of the Gram matrix on all $\mathbf{x}_i, \mathbf{y}_j$.

$$\begin{bmatrix} \alpha^T & \mathbf{0}_n^T \\ \mathbf{0}_m^T & \beta^T \end{bmatrix} \begin{bmatrix} K_{\mathbf{x}} & K_{\mathbf{x},\mathbf{y}} \\ K_{\mathbf{x},\mathbf{y}}^T & K_{\mathbf{y}} \end{bmatrix} \begin{bmatrix} \alpha & \mathbf{0}_m \\ \mathbf{0}_n & \beta \end{bmatrix} = \begin{bmatrix} \alpha^T K_{\mathbf{x}} \alpha & \alpha^T K_{\mathbf{x},\mathbf{y}} \beta \\ \beta^T K_{\mathbf{x},\mathbf{y}}^T \alpha & \beta^T K_{\mathbf{y}} \beta \end{bmatrix} \succeq 0$$

hence

$$\|f\|^2 \|g\|^2 = (\alpha^T K_{\mathbf{x}} \alpha)(\beta^T K_{\mathbf{y}}) \geq (\alpha^T K_{\mathbf{x},\mathbf{y}} \beta)^2 = \langle f, g \rangle^2.$$

- Hence $\|f\| = 0 \Rightarrow f = 0$ since

$$\forall \mathbf{x} \in \mathcal{X}, |f(\mathbf{x})| = \langle f, k(\mathbf{x}, \cdot) \rangle \leq \|f\| \sqrt{k(\mathbf{x}, \mathbf{x})} = 0.$$

- $\tilde{\mathcal{H}}$ is a pre-Hilbertian. For any Cauchy sequence f_n in $\tilde{\mathcal{H}}$, and $\mathbf{x} \in \mathcal{X}$

$$|f_m(\mathbf{x}) - f_n(\mathbf{x})| = \langle f_n - f_m, k(\mathbf{x}, \cdot) \rangle \leq \|f_n - f_m\| \sqrt{k(\mathbf{x}, \mathbf{x})} \to 0,$$

  $f_n(\mathbf{x})$ is thus Cauchy in $\mathbb{R}$ and has thus a limit. f_n has thus a limit.
- We add all such limits to **complete** $\tilde{\mathcal{H}}$ into $\mathcal{H}$.
- still a few steps more (show the r.k. of $\mathcal{H}$ is still k).

Another alternative definition

Definition 4 (Reproducing Kernel)**.** *A real-valued function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a reproducing kernel of a Hilbert space* $\mathcal{H}$ *of real-valued functions on* $\mathcal{X}$ *if and only if*

- $\forall t \in \mathcal{X}, \;\; k(\cdot, t) \in \mathcal{H};$
- $\forall t \in \mathcal{X}, \forall f \in \mathcal{H}, \;\; \langle f, k(\cdot, t) \rangle = f(t).$

- straightforward to prove equivalence with the first characterization.

A more intuitive perspective: Feature maps

Theorem 2. *A function k on $\mathcal{X} \times \mathcal{X}$ is a positive definite kernel if and only if there exists a set T and a mapping ϕ from $\mathcal{X}$ to $l^2(T)$, the set of real sequences $\{u_t, t \in T\}$ such that $\sum_{t\in T} |u_t|^2 < \infty$, where*

$$\forall(\mathbf{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{X}\,,\; k(\mathbf{x}, \boldsymbol{y}) = \sum_{t\in T} \phi(\mathbf{x})_t\, \phi(\boldsymbol{y})_t = \langle \phi(\mathbf{x}), \phi(\boldsymbol{y}) \rangle_{l^2(T)}$$

- A very popular perspective in the machine learning world.
- Equivalent to previous definitions, less stressed in the RHKS literature.

$$\mathbf{x} \longrightarrow \phi(\mathbf{x}) = \begin{bmatrix} \vdots \\ \vdots \\ \phi(\mathbf{x})_t \\ \vdots \\ \vdots \end{bmatrix}_{t\in T}$$

where the ϕ_t are a set of **possibly infinite** but countable features.

positive definite kernels and distances

- Kernels are often called similarities.
- the **higher** $k(\mathbf{x}, \mathbf{y})$, the more similar $\mathbf{x}$ and $\mathbf{y}$.
- With distances, the **lower** $d(\mathbf{x}, \mathbf{y})$, the closer $\mathbf{x}$ and $\mathbf{y}$.

- Many distances exist in the literature. Can they be used to define kernels?

what is the link between kernels and distances?
high similarity $\stackrel{?}{=}$ **small distance**

- At least true for the Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$...
- Important theorems taken from [BCR84].

Distances

Definition 5 (Distances, or metrics). *A* ***nonnegative-valued*** *function d on $\mathcal{X} \times \mathcal{X}$ is a distance if it satisfies, $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$:*

- $d(\mathbf{x}, \mathbf{y}) \geq 0$, *and* $d(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$ *(non-degeneracy)*
- $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ *(symmetry),*
- $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ *(triangle inequality)*

- Very simple example: if $\mathcal{X}$ is a Hilbert space, $\|\mathbf{x} - \mathbf{y}\|$ is a distance. It is usually called a... **Hilbertian distance**.
- By extension, any distance $d(\mathbf{x}, \mathbf{y})$ which can be written as $\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|$ where ϕ maps $\mathcal{X}$ to any Hilbert space is called a **Hilbertian metric**.
- Useful. To build Gaussian kernel, Laplace kernels $k(\mathbf{x}, \mathbf{y}) = e^{-t\|\mathbf{x}-\mathbf{y}\|}$...
- Yet this concept is a bit too restrictive and does not contain all interesting distances.

the missing link: negative definite kernels

Definition 6 (Negative Definite Kernels)**.** *A symmetric function* $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a negative definite (n.d.) kernel on* $\mathcal{X}$ *if*

$$\sum_{i,j=1}^{n} c_i c_j \, \psi\left(x_i, x_j\right) \leq 0 \tag{1}$$

holds for any $n \in \mathbb{N}, x_1, \ldots, x_n \in \mathcal{X}$ *and* $c_1 \ldots, c_n \in \mathbb{R}$ *such that* $\sum_{i=1}^{n} c_i = 0$.

- Example $\psi(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$.
  - prove by decomposing into $\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle$
- $\mathcal{N}(\mathcal{X})$ is also a closed convex cone.

important example: k is p.d. $\Rightarrow -k$ is n.d.
Converse completely false.

negative definite kernels & positive definite kernels

A first link between these two kernels:

Proposition 3. *Let* $x_0 \in \mathcal{X}$ *and let* $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *be a symmetric kernel. Let*

$$\varphi(\mathbf{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \psi(\mathbf{x}, x_0) + \psi(\boldsymbol{y}, x_0) - \psi(\mathbf{x}, \boldsymbol{y}) - \psi(x_0, x_0).$$

Then k *is positive definite* $\Leftrightarrow \psi$ *is negative definite.*

- Example: $\|\mathbf{x} - x_0\|^2 + \|\mathbf{y} - x_0\|^2 - \|\mathbf{x} - \mathbf{y}\|^2$ is a p.d. kernel.

Proof.

- $\Rightarrow$ For $\mathbf{x}_1, \cdots, \mathbf{x}_n$, and $c_1, \cdots, c_n$ s.t. $\sum_{i=1}^n c_i = \mathbf{0}$,

$$\sum_{i,j=1}^{n} c_i c_j \varphi(\mathbf{x}_i, \mathbf{x}_j) = -\sum_{i,j=1}^{n} c_i c_j \psi(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

- $\Leftarrow$ For $\mathbf{x}_1, \cdots, \mathbf{x}_n$ and $c_1, \cdots, c_n$, let $c_0 = -\sum_{i=1}^n$. Set $\mathbf{x}_0 = x_0$. Then

$$\begin{aligned} 0 &\geq \sum_{i,j=0}^{n} c_i c_j \psi(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{i,j=1}^{n} c_i c_j \psi(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n} c_i c_0 \psi(\mathbf{x}_i, x_0) + \sum_{j=1}^{n} c_0 c_j \psi(x_0, \mathbf{x}_j) + c_0^2 \psi(x_0, x_0). \\ &= \sum_{i,j=1}^{n} [\psi(\mathbf{x}_i, x_0) + \psi(\mathbf{x}_j, x_0) - \psi(\mathbf{x}_i, \mathbf{y}_j) - \psi(x_0, x_0)] = \sum_{i,j=1}^{n} c_i c_j \varphi(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

Proposition 4. *For a p.d. kernel $k \geq 0$ on $\mathcal{X} \times \mathcal{X}$, the following conditions are equivalent*

- $-\log k \in \mathcal{N}(\mathcal{X})$,
- k^t *is positive definite for all* $t > 0$.

If k satisfies either, k is said to be ***infinitely divisible***,

Proof.

- $-\log k = \lim_{n\to\infty} n(1 - k^{\frac{1}{n}})$ which is the limit of a series of n.d. kernels if (ii) is true, hence $(ii) \Rightarrow (i)$.
- conversely, if $-\log k \in \mathcal{N}(\mathcal{X})$ we use Proposition 3. Writing $\psi = -\log k$ and choosing $x_0 \in \mathcal{X}$ we have

$$k^t = e^{-t\psi(\mathbf{x},\mathbf{y})} = e^{t\psi(x_0,x_0)} e^{t\varphi(\mathbf{x},\mathbf{y})} e^{-t\psi(\mathbf{x},x_0)} e^{-t\psi(\mathbf{y},x_0)} \in \mathcal{P}(\mathcal{X})$$

negative definite kernels: $(\text{Hilbertian distance})^2$ + ...

Proposition 5. *Let $\psi : \mathcal{X} \times \mathcal{X}$ be a n.d. kernel. Then there is a Hilbert space H and a mapping ϕ from X to H such that*

$$\psi(\boldsymbol{x}, \boldsymbol{y}) = \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})\|^2 + f(\boldsymbol{x}) + f(\boldsymbol{y}), \tag{2}$$

where $f : \mathcal{X} \to \mathbb{R}$. If $\psi(x, x) = 0$ for all $\mathbf{x} \in \mathcal{X}$ then f can be chosen as zero. If the set $\{(\boldsymbol{x}, \boldsymbol{y}) | \, \psi(\boldsymbol{x}, \boldsymbol{y}) = 0\}$ is exactly $\{(\mathbf{x}, \mathbf{x}), \mathbf{x} \in \mathcal{X}\}$ then $\sqrt{\psi}$ is a Hilbertian distance.

Proof. Fix x_0 and define

$$\varphi(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \frac{\mathbf{1}}{\mathbf{2}} \left[\psi(\mathbf{x}, x_0) + \psi(\mathbf{y}, x_0) - \psi(\mathbf{x}, \mathbf{y}) - \psi(x_0, x_0) \right].$$

By Proposition 3 φ is p.d. hence there is a RKHS and mapping ϕ such that $\varphi(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$. Hence

$$\begin{aligned} \|\phi(\mathbf{x}) - \phi(\mathbf{y})\|^2 &= \varphi(\mathbf{x}, \mathbf{x}) + \varphi(\mathbf{y}, \mathbf{y}) - 2\varphi(\mathbf{x}, \mathbf{y}) \\ &= \psi(\mathbf{x}, \mathbf{y}) - \frac{\psi(\mathbf{x}, \mathbf{x}) + \psi(\mathbf{y}, \mathbf{y})}{2}. \end{aligned}$$

distances & negative definite kernels

- whenever a n.d. kernel ψ
  - vanishes on the *diagonal*, $i.e.$ on $\{(x,x), x \in \mathcal{X}\}$,
  - is 0 only on the diagonal, to ensure non-degeneracy,

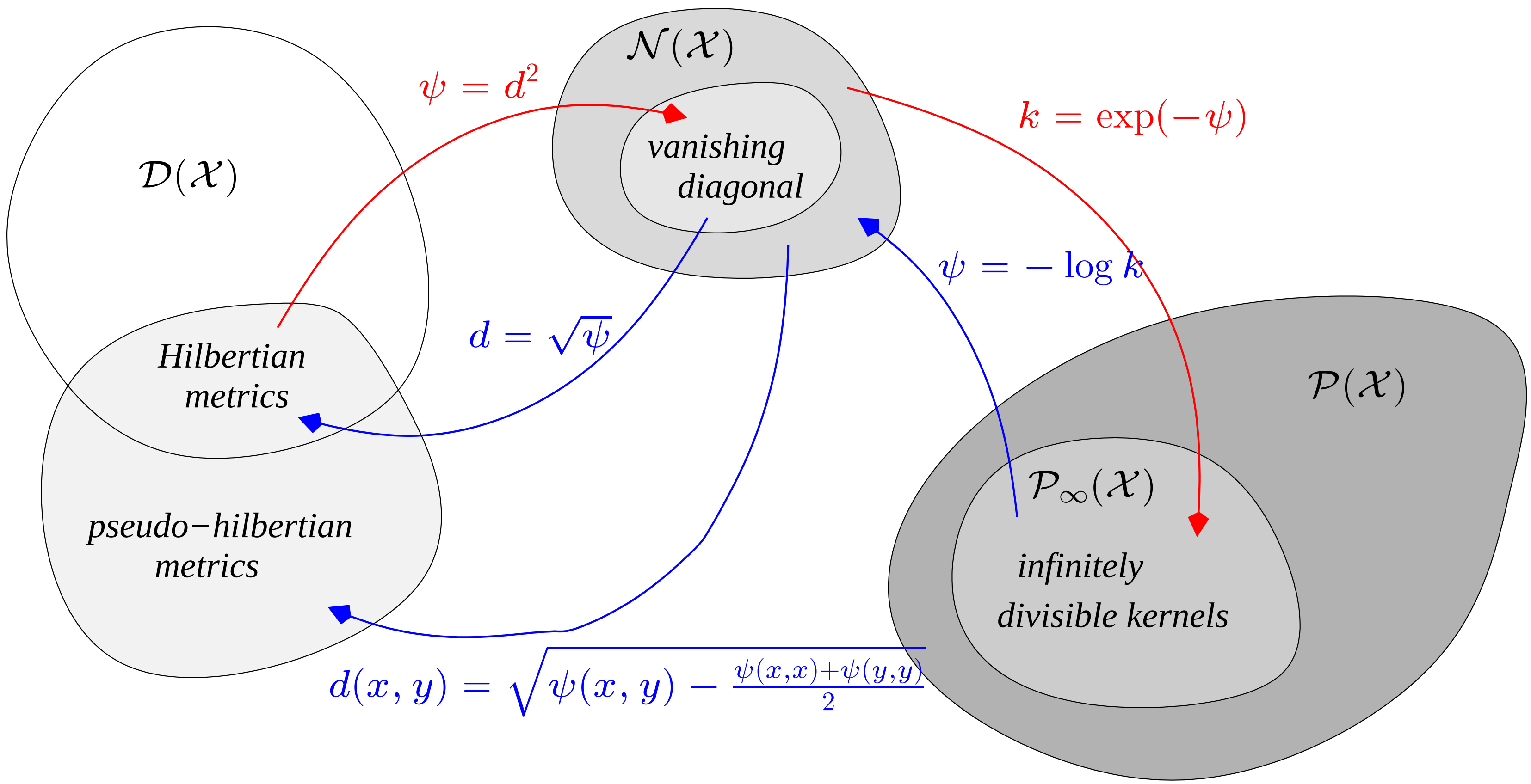
$$\rightarrow \sqrt{\psi} \text{ is a Hilbertian distance for } \mathcal{X}.$$

- **More generally**, for a n.d. kernel ψ,

$$\sqrt{\psi(\mathbf{x},\mathbf{y}) - \frac{\psi(\mathbf{x},\mathbf{x})}{2} - \frac{\psi(\mathbf{y},\mathbf{y})}{2}} \text{ is a (pseudo)}\textbf{metric}\text{ for } \mathcal{X}\ .$$

- On the contrary, to each distance does not always correspond a n.d. kernel (Monge-Kantorovich distance, edit-distance $etc.$.)

In summary...



- Set of distances on $\mathcal{X}$ is $\mathcal{D}(\mathcal{X})$, Negative definite kernels $\mathcal{N}(\mathcal{X})$, positive and infinitely divisible positive kernels $\mathcal{P}(\mathcal{X})$ and $\mathcal{P}_\infty(\mathcal{X})$ respectively.

Some final remarks on $\mathcal{N}(\mathcal{X})$ and $\mathcal{P}(\mathcal{X})$

- $\mathcal{N}(\mathcal{X})$ is a cone. Additionally,
  - if $\psi \in \mathcal{N}(\mathcal{X}), \forall c \in \mathbb{R}$, $\psi + c \in \mathcal{N}(\mathcal{X})$.
  - if $\psi(x,x) \geq 0$ for all $x \in \mathcal{X}$, $\boldsymbol{\psi^\alpha \in \mathcal{N}(\mathcal{X})}$ for $0 < \alpha < 1$ since

$$\psi^\alpha = \frac{\alpha}{\Gamma(1-\alpha)} \int_0^\infty t^{-\alpha-1}(1-e^{-t\psi})dt$$

  and $\boldsymbol{\log(1+\psi) \in \mathcal{N}(\mathcal{X})}$ since

$$\log(1+\psi) = \int_0^\infty (1-e^{-t\psi})\frac{e^{-t}}{t}dt.$$

  - if $\psi > 0$, then $\boldsymbol{\log(\psi) \in \mathcal{N}(\mathcal{X})}$ since

$$\log(\psi) = \lim_{c\to\infty} \log\left(\psi + \frac{1}{c}\right) = \lim_{c\to\infty} \log\left(1 + c\psi\right) - \log c$$

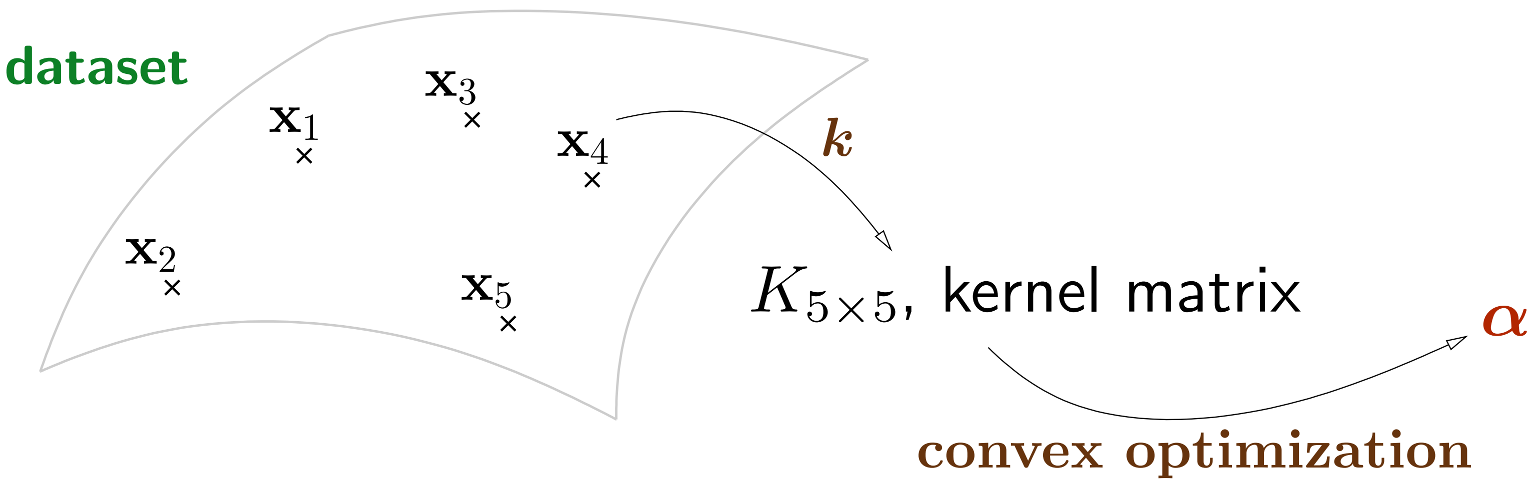
Some final remarks on $\mathcal{D}(\mathcal{X}), \mathcal{N}(\mathcal{X}), \mathcal{P}(\mathcal{X})$

- $\mathcal{P}(\mathcal{X})$ is a cone. Additionally,
  - The pointwise product $k_1 k_2$ of two p.d. kernels if a p.d. kernel
  - $k^n \in \mathcal{P}(\mathcal{X})$ for $n \in \mathbb{N}$. $(k+c)^n$ too...as well as $\exp(k) \in \mathcal{P}(\mathcal{X})$:
    - $\exp(k) = \sum_{i=0}^{\infty} \frac{k^i}{i!}$, a limit of p.d. kernels.
    - $\exp(k) = \exp(-(-k))$ where $-k \in \mathcal{N}(\mathcal{X})$.
- The sum of two infinitely divisible kernels is not necessarily infinitely divisible.
  - $-\log k_1$ and $-\log k_2$ might be in $\mathcal{N}(\mathcal{X})$, but $-\log(k_1 + k_2)$?...

Defining kernels

Intuitively an important issue...

Remember that kernel methods drop all previous information



to proceed exclusively with K.

if the kernel K is poorly informative, the optimization cannot be very useful...
it is therefore **crucial** that the kernel quantifies **noteworthy similarities**.

Kernels on vectors

(relatively) easy case: **we are only given feature vectors**,
with **no access** to the original data.

- Reminder (copy paste of previous slide!): for a family of kernels $k_1, \cdots, k_n, \cdots$
  - The sum $\sum_{i=1}^{n} \lambda_i k_i$ is p.d., given $\lambda_1, \ldots, \lambda_n \geq 0$
  - The product $k_1^{a_1} \cdots k_n^{a_n}$ is p.d., given $a_1, \ldots, a_n \in \mathbb{N}$
  - $\lim_{n\to\infty} k_n$ is p.d. (if the limit exists!).
- Using these properties we can prove the p.d. of
  - the polynomial kernel $k_p(x,y) = (\langle \mathbf{x}, \mathbf{y} \rangle + b)^d, \quad b > 0, d \in \mathbb{N}$,
  - the Gaussian kernel $k_\sigma(x,y) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$ which can be rewritten as

$$k_\sigma(x,y) = \left[e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} e^{-\frac{\|\mathbf{y}\|^2}{2\sigma^2}} \right] \cdot \left[\sum_{i=0}^{\infty} \frac{\langle \mathbf{x}, \mathbf{y} \rangle^i}{i!} \right]$$

Kernels on vectors

- the Laplace kernels, using some n.d. kernel weaponry,

$$k_\lambda(x,y) = e^{-\lambda\|\mathbf{x}-\mathbf{y}\|^{\boldsymbol{a}}}, \quad 0 < \lambda,\, 0 < \boldsymbol{a} \leq 2$$

- the all-subset Gaussian kernel in $\mathbb{R}^d$,

$$k(x,y) = \prod_{i=1}^{d}\left(1 + ae^{-b(x_i-y_i)^2}\right) = \sum_{I\subset\{1,\cdots,d\}} a^{\#(I)}e^{-b\|\mathbf{x}_I-\mathbf{y}_I\|^2}.$$

- A variation on the Gaussian kernel: Mahalanobis kernel,

$$k_\Sigma(x,y) = e^{-(\mathbf{x}-\mathbf{y})^T\Sigma^{-1}(\mathbf{x}-\mathbf{y})},$$

  idea: correct for discrepancies between the magnitudes and correlations of different variables.
- Usually Σ is the empirical covariance matrix of a sample of points.

Kernels on vectors

- These kernels can be seen as *meta*-kernels which can use any feature representation.
- Example: Gaussian kernel of Gaussian kernel feature maps,

$$k_{G^2}(\mathbf{x}, \mathbf{y}) = k_G\left(e^{-\frac{\|\mathbf{x}-\cdot\|^2}{2\sigma^2}}, e^{-\frac{\|\mathbf{y}-\cdot\|^2}{2\sigma^2}}\right) = e^{-\frac{2-e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}}{2\lambda^2}}.$$

- Not sure this is very useful though!
- Indeed, the real challenge is not to define funky kernels,

the challenge is to tune the parameters b, d, σ, Σ.

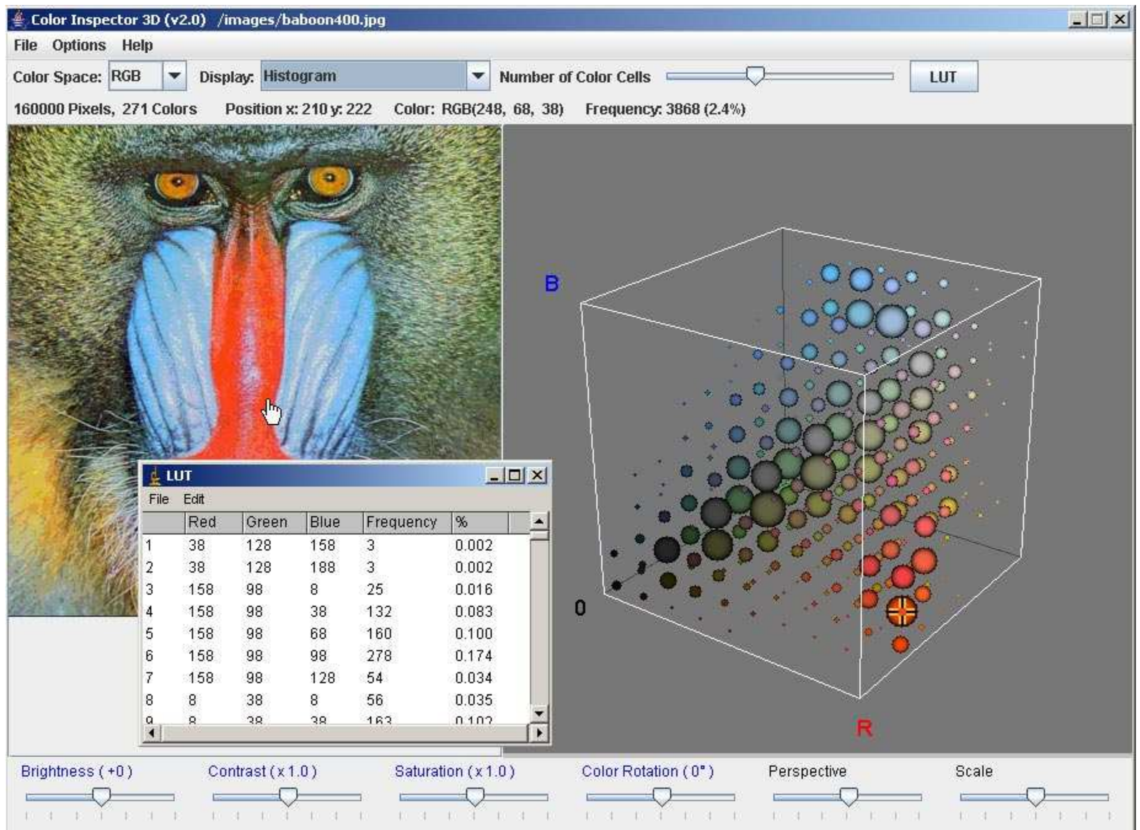
Kernels on structured objects

- Structured objects?
  - texts, webpages, documents
  - sounds, speech, music,
  - images, video segments, movies,
  - 3d structures, sequences, trees, graphs
- Structured objects means
  - objects with **a tricky structure**,
  - which cannot be simply embedded in a vector space of small dimensionality,
  - without obvious algebraic properties,

structured object = that which cannot be represented in a (small) Euclidian space

Vectors in $\mathbb{R}^n_+$ and Histograms

- A powerful and popular feature representation for structured objects: **histograms of smaller building-blocks of the object**:



- histograms are simple instances of **probability measures**,
  - nonnegative coordinates, sum up to 1.

Standard metrics for Histograms

Information geometry, introduced yesterday, studies distances between densities.

- Reference : [AN01]
- An abridged bestiary of **negative definite distances** on the probability simplex:

$$\psi_{JD}(\theta,\theta') = h\left(\frac{\theta+\theta'}{2}\right) - \frac{h(\theta)+h(\theta')}{2},$$

$$\psi_{\chi^2}(\theta,\theta') = \sum_i \frac{(\theta_i-\theta_i')^2}{\theta_i+\theta_i'}, \quad \psi_{TV}(\theta,\theta') = \sum_i |\theta_i-\theta_i'|,$$

$$\psi_{H_2}(\theta,\theta') = \sum_i |\sqrt{\theta_i}-\sqrt{\theta_i'}|^2, \quad \psi_{H_1}(\theta,\theta') = \sum_i |\sqrt{\theta_i}-\sqrt{\theta_i'}|.$$

- Recover kernels through

$$k(\theta,\theta') = e^{-t\psi}, \quad t>0$$

Information Diffusion Kernel [LL05,ZLC05]

- Solve the heat equation on the multinomial manifold, using the Fisher metric
- Approximate the solution with

$$k_{\Sigma_d}(\theta, \theta') = e^{-\frac{1}{t}\arccos^2(\sqrt{\theta}\cdot\sqrt{\theta'})},$$

- $\arccos^2$ is the **squared** **geodesic distance** between θ and θ' as elements from the unit sphere ($\theta_i \to \sqrt{\theta_i}$).
- In [ZLC05]: the use of

$$k_{\Sigma_d}(\theta, \theta') = e^{-\frac{1}{t}\arccos(\sqrt{\theta}\cdot\sqrt{\theta'})},$$

  is advocated.
- the geodesic distance is a n.d. kernel on the *whole sphere* ($\arccos^2$ is not).

Statistical Modeling and Kernels

Histograms cannot always summarize efficiently the structures of $\mathcal{X}$

- Statistical models of complex objects provide richer explanations:
  - Hidden Markov Models for sequences and time-series,
  - VAR, VARMA, ARIMA *etc.* models for time-series,
  - Branching processes for trees and graphs
  - Random Markov Fields for images *etc.*
- $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ are interpreted as i.i.d realizations of one or many densities on $\mathcal{X}$.
- These densities belong to a model $\{p_{\boldsymbol\theta}, \boldsymbol\theta \in \boldsymbol\Theta \subset \mathbb{R}^d\}$

Can we use **generative** (statistical) **models** in **discriminative** (kernel and metric based) **methods**?

Fisher Kernel

- The Fisher kernel [JH99] between two elements $\mathbf{x}, \mathbf{y}$ of $\mathcal{X}$ is

$$k_{\hat{\theta}}(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial \ln \boldsymbol{p_\theta}(\mathbf{x})}{\partial \theta} \Big|_{\hat{\boldsymbol{\theta}}} \right)^T \boldsymbol{J}_{\hat{\boldsymbol{\theta}}}^{-1} \left(\frac{\partial \ln \boldsymbol{p_\theta}(\mathbf{y})}{\partial \theta} \Big|_{\hat{\boldsymbol{\theta}}} \right),$$

  - $\hat{\boldsymbol{\theta}}$ has been selected using sample data ($e.g.$ MLE),
  - $\boldsymbol{J}_{\hat{\boldsymbol{\theta}}}^{-1}$ is the Fisher information matrix computed in $\hat{\theta}$.

- The statistical model $\{\boldsymbol{p_\theta}, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ provides:
  - finite dimensional *features* through the **score vectors**,
  - A ***Mahalanobis metric*** associated with these vectors through $J_{\hat{\theta}}$.

- Alternative formulation:

$$k_{\hat{\theta}}(x, y) = e^{-\frac{1}{\sigma^2}\left(\nabla_{\hat{\theta}} \ln p_\theta(\mathbf{x}) - \nabla_{\hat{\theta}} \ln p_\theta(\mathbf{y})\right)^T J_{\hat{\theta}}^{-1}\left(\nabla_{\hat{\theta}} \ln p_\theta(\mathbf{x}) - \nabla_{\hat{\theta}} \ln p_\theta(\mathbf{y})\right)}.$$

with the meta-kernel idea.

Fisher Kernel Extended [TKR+02,SG02]

- Minor extensions, useful for binary classification:
- Estimate $\hat{\theta}_1$ and $\hat{\theta}_2$ for each class respectively,
- consider the score vector of the likelihood ratio

$$\phi_{\hat{\theta}_1,\hat{\theta}_2} : \mathbf{x} \mapsto \left(\frac{\partial \ln \frac{p_{\theta_1}(\mathbf{x})}{p_{\theta_2}(\mathbf{x})}}{\partial \vartheta} \bigg|_{\hat{\vartheta}=(\hat{\theta}_1,\hat{\theta}_2)} \right),$$

where $\vartheta = (\theta_1, \theta_2)$ is in Θ^2.

- Use this logratio's score vector to propose instead the kernel

$$(x, y) \mapsto \phi_{\hat{\theta}_1,\hat{\theta}_2}(\mathbf{x})^T \phi_{\hat{\theta}_1,\hat{\theta}_2}(\mathbf{y}).$$

Mutual Information Kernel: densities as feature extractors

- More **bayesian** flavor $\rightarrow$ drops maximum-likelihood estimation of θ. [See02]
- Instead, use **prior knowledge** on $\boldsymbol{\{p_\theta, \theta \in \Theta\}}$ through a **density** ω on Θ
- Mutual information kernel k_ω:

$$k_\omega(\mathbf{x}, \mathbf{y}) = \int_\Theta p_\theta(\mathbf{x}) p_\theta(\mathbf{y})\, \omega(d\theta).$$

- The feature maps $0 \leq p_\theta(\mathbf{x}) \leq 1$ and $0 \leq p_\theta(\mathbf{y}) \leq 1$.

> k_ω is big whenever many **common** densities p_θ
> score high probabilities for **both** $\mathbf{x}$ and $\mathbf{y}$

- Explicit computations sometimes possible, **namely conjugate priors**.
- Example: context-tree kernel for strings.

Mutual Information Kernel & Fisher Kernels

The Fisher kernel is a maximum *a posteriori* approximation of the MI kernel.

- What? How? by setting the prior ω to the multivariate Gaussian density

$$\mathcal{N}(\hat{\theta}, J_{\hat{\theta}}^{-1}),$$

  an approximation known as Laplace's method,

- Writing

$$\Phi(x) = \nabla_{\hat{\theta}} \ln p_\theta(x) = \frac{\partial \ln p_\theta(x)}{\partial \theta}\Big|_{\hat{\theta}}$$

  we get

$$\log p_\theta(x) \approx \log p_{\hat{\theta}}(x) + \Phi(x)(\theta - \hat{\theta}).$$

Mutual Information Kernel & Fisher Kernels

- Using $\mathcal{N}(\hat{\theta}, J_{\hat{\theta}}^{-1})$ for ω yields

$$
\begin{aligned}
k(x,y) &= \int_\Theta p_\theta(\mathbf{x}) p_\theta(\mathbf{y})\, \omega(d\theta), \\
&\approx C \int_\Theta e^{\log p_{\hat{\theta}}(x) + \Phi(x)^T(\theta - \hat{\theta})} e^{\log p_{\hat{\theta}}(y) + \Phi(y)^T(\theta - \hat{\theta})} \; e^{-(\theta-\hat{\theta})^T J_{\hat{\theta}}(\theta - \hat{\theta})} d\theta \\
&= C p_{\hat{\theta}}(x) p_{\hat{\theta}}(y) \int_\Theta e^{(\Phi(x)+\Phi(y))^T(\theta-\hat{\theta}) + (\theta-\hat{\theta})^T J_{\hat{\theta}}(\theta-\hat{\theta})} d\theta \\
&= C' p_{\hat{\theta}}(x) p_{\hat{\theta}}(y) e^{\frac{1}{2}(\Phi(x)+\Phi(y))^T J_{\hat{\theta}}^{-1}(\Phi(x)+\Phi(y))}
\end{aligned}
\tag{1}
$$

- the kernel

$$\tilde{k}(x,y) = \frac{k(x,y)}{\sqrt{k(x,x)k(y,y)}}$$

is equal to the Fisher kernel in exponential form.

Marginalized kernels - Graphs and Sequences

- Similar ideas: leverage **latent variable models.** [TKA02,KTI03]
- For **location** or **time-based** data,
  - the probability of emission of a token x_i is conditioned by
  - an **unobserved** latent variable $s_i \in \mathcal{S}$, where $\mathcal{S}$ is a finite space of possible states.
- for observed sequences $\mathbf{x} = (x_1, \cdots, x_n), \mathbf{y} = (y_1, \cdots, y_n)$, sum over all possible state sequences the **weighted** product of **these probabilities**:

$$k(x, y) = \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} \boldsymbol{p(s|x)}\, \boldsymbol{p(s'|y)}\, \boldsymbol{\kappa}\left((\boldsymbol{x}, \boldsymbol{s}), (\boldsymbol{y}, \boldsymbol{s'})\right)$$

- closed form computations exist for graphs & sequences.

Kernels on MLE parameters

- Use model directly to extract a single representation from observed points:

$$x \mapsto \hat{\theta}_x, \quad y \mapsto \hat{\theta}_y,$$

through MLE for instance.

- compare **x** and **y** through a kernel k_Θ on Θ,

$$k(x, y) = k_\Theta(\hat{\theta}_{\mathbf{x}}, \hat{\theta}_{\mathbf{y}}).$$

- Bhattacharrya affinities:

$$k_\beta(\mathbf{x}, \mathbf{y}) = \int_{\mathcal{X}} p_{\hat{\theta}_{\mathbf{x}}}(z)^\beta p_{\hat{\theta}_{\mathbf{y}}}(z)^\beta dz$$

for $\beta > 0$.