EPAT 2010

Kernel Methods

Algorithms

Marco Cuturi

Outline of the lectures

Outline

- Mathematical considerations $(\leq 80's)$
  - Reproducing Kernel Hilbert Spaces
  - positive-definiteness, negative definiteness $etc.$.
  - kernels, similarities and distances
- Defining kernels
  - Standard kernels $(\leq 80's)$
  - Statistical modeling & kernels (> 1998)
  - Algebraic structures and kernels
- Kernel algorithms
  - supervised learning, SVM (≥ 1995)
  - representer theorem
  - unsupervised techniques, eigenfunctions of samples (≥ 1998)
  - density estimation and novelty detection (≥ 1999)

Kernel algorithms

algorithms which select functions with desirable properties **in a RKHS**
algorithms which only take as **inputs Gram matrices** K

Regression, Classification and other Supervised Tasks

- **Two associated random variables**
  - A random variable x, taking values in $\mathcal{X}$,
  - A random variable y, taking values in $\mathcal{Y}$.
- **Two samples of (x, y) i.i.d. distributed from their joint law**
  - $\{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$, n couples of $\mathcal{X} \times \mathcal{Y}$.

Challenge: **predict** $\mathbf{y}$ when given only $\mathbf{x}$.

- In practice, **find** a function $\mathcal{X} \to \mathcal{Y}$ for which $f(\mathbf{x})$ is not too different from y on average.

Binary Classification

- $\mathcal{Y} = -1, 1.$
- f needs to be a functions that, given $\mathbf{x}$ predicts a label,

$$f : \mathcal{X} \mapsto \{0, 1\}$$

of course, many possible choices for f's shape.

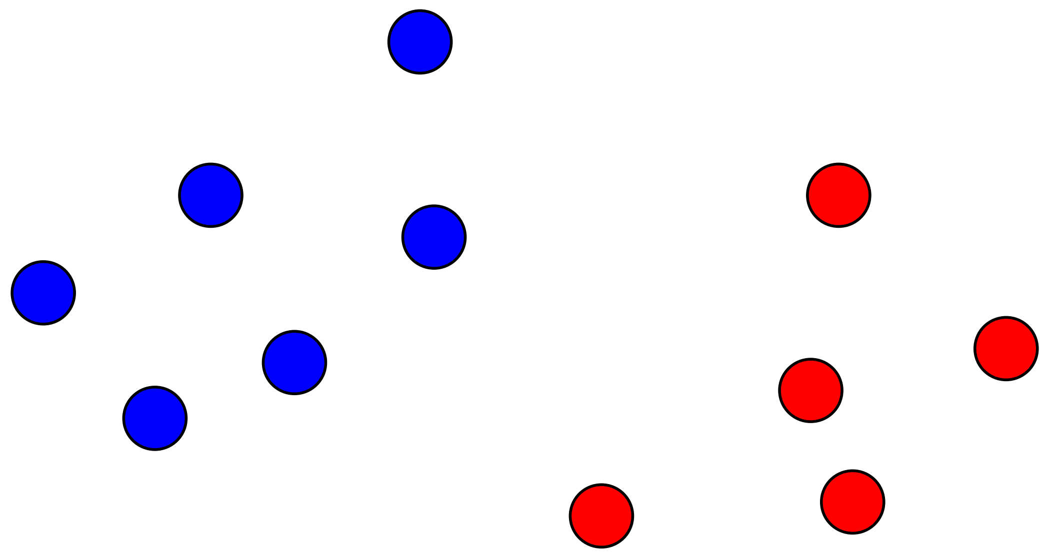
- We review here **linear** hyperplanes in $\mathcal{X} = \mathbb{R}^d$ first.
- We represent it in $\mathbb{R}^2$ for simplicity.

Next slides will cover an important algorithm, the **SVM** algorithm

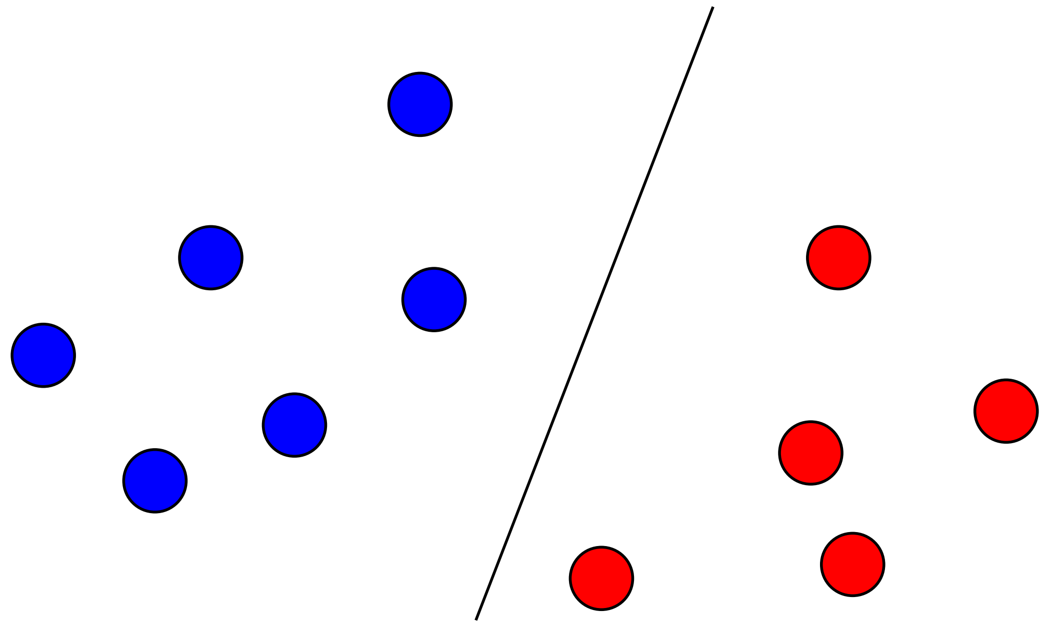
- this algorithm can be naturally expressed in terms of *kernels*. we review later other algorithms for which this is also the case.

thanks to Jean-Philippe Vert for many of the following figures and slides.

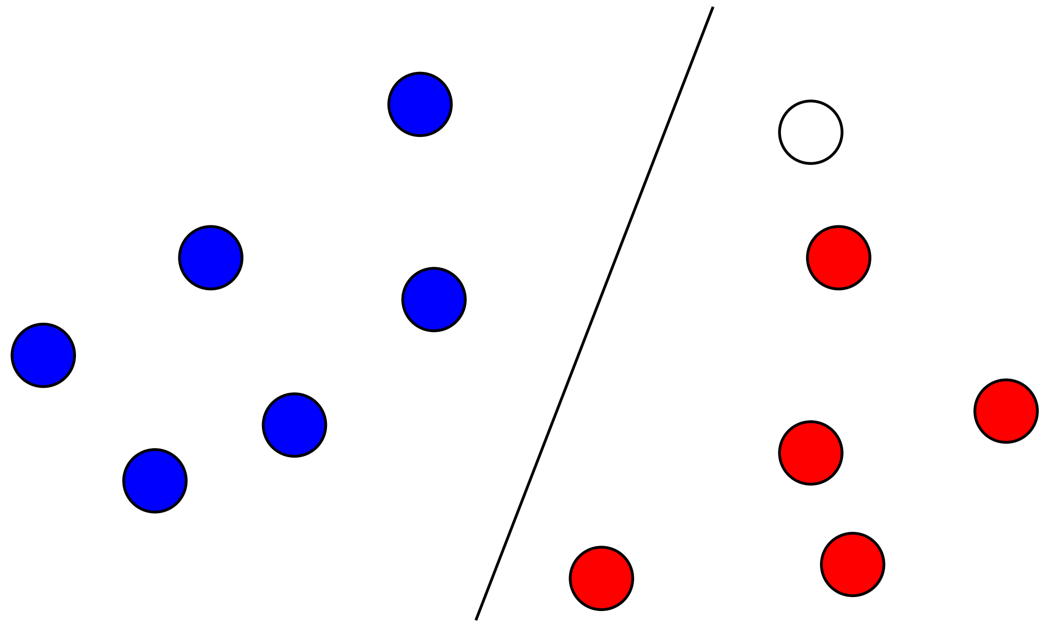
Linear classifier, some degrees of freedom



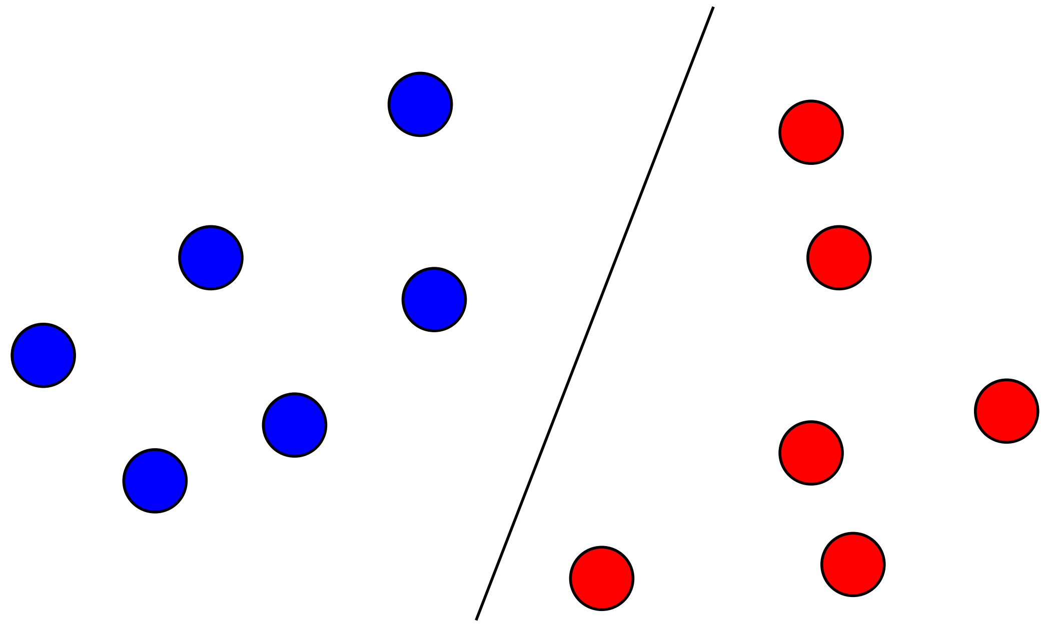
Linear classifier, some degrees of freedom



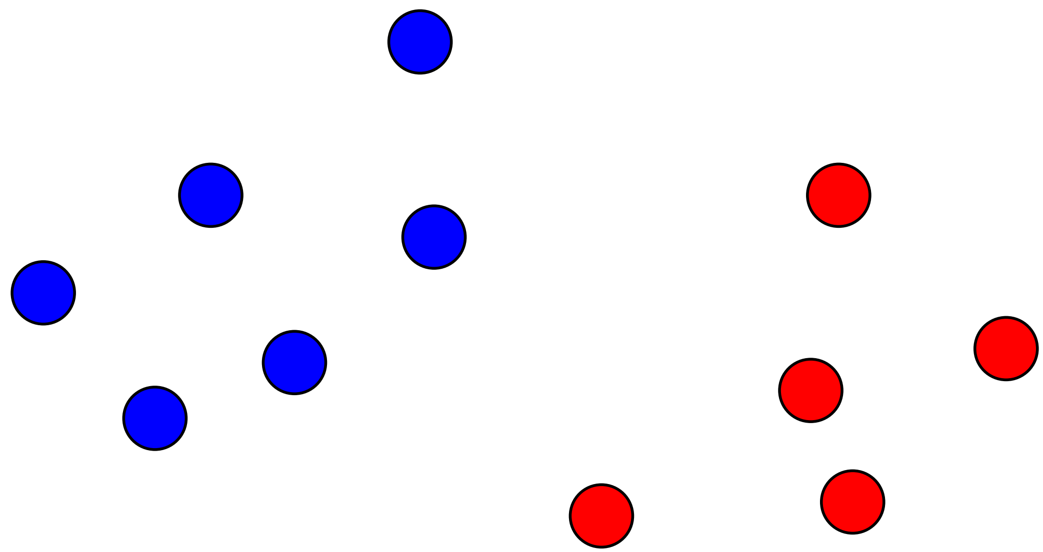
Linear classifier, some degrees of freedom



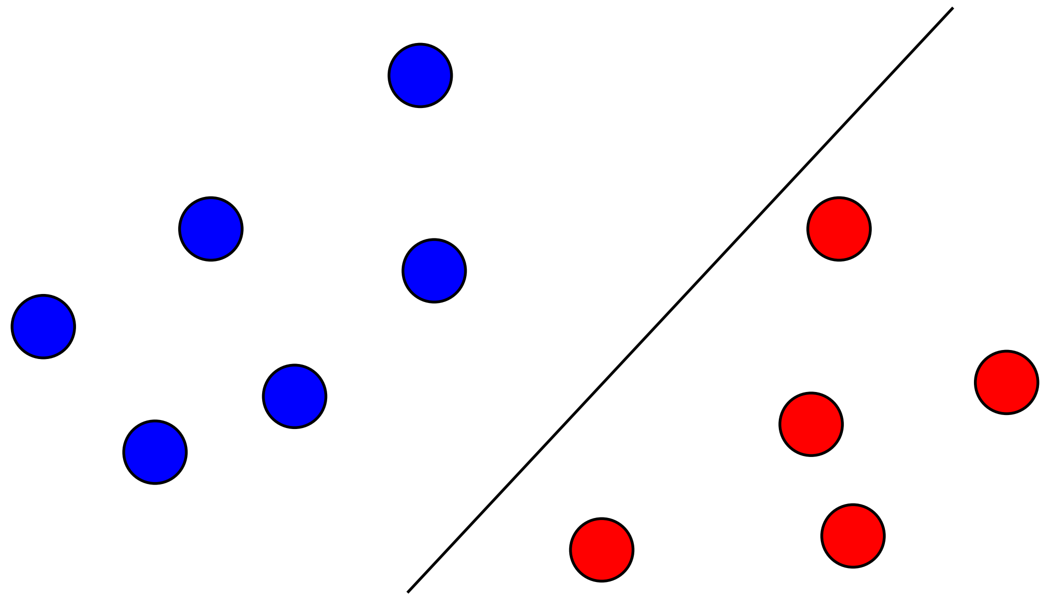
Linear classifier, some degrees of freedom



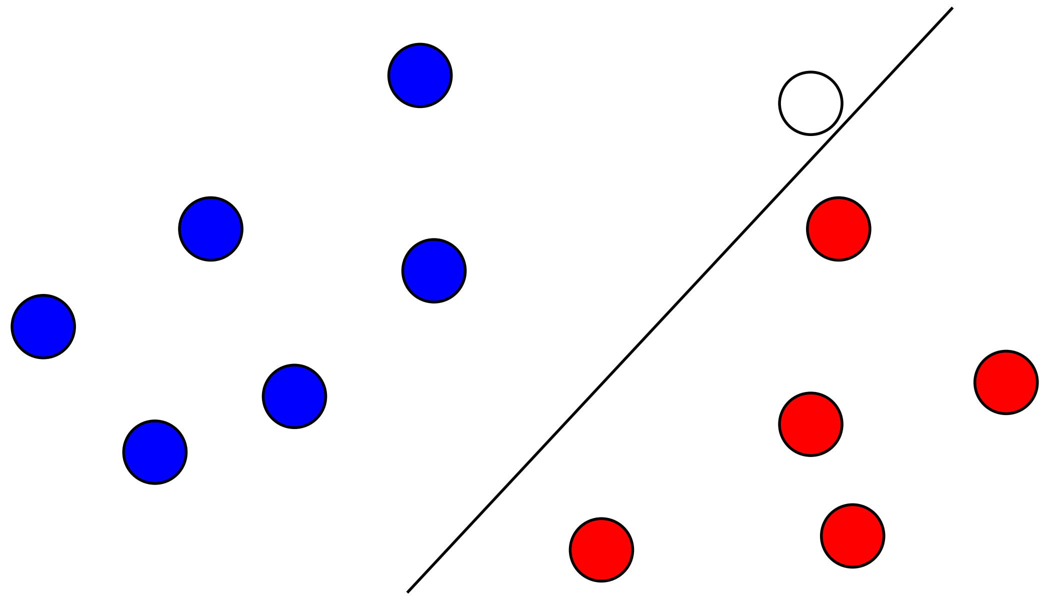
Linear classifier, some degrees of freedom



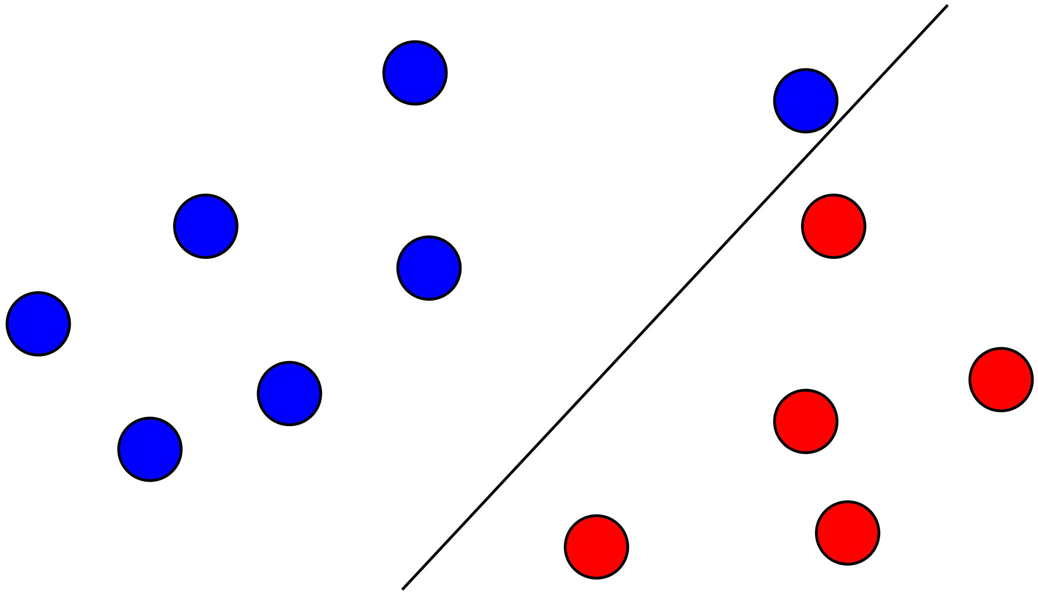
Linear classifier, some degrees of freedom



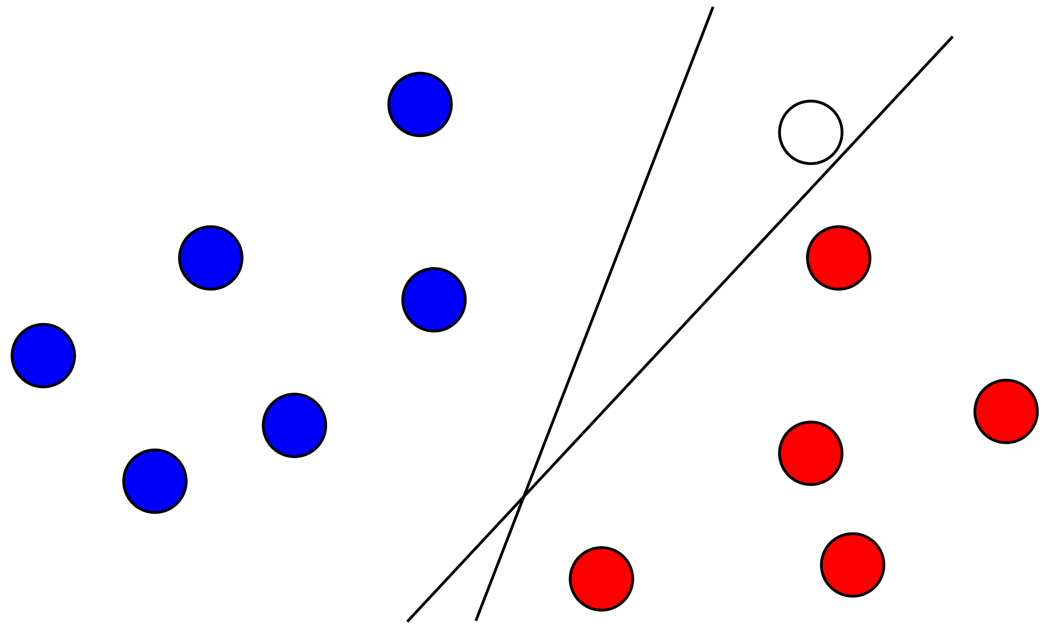
Linear classifier, some degrees of freedom



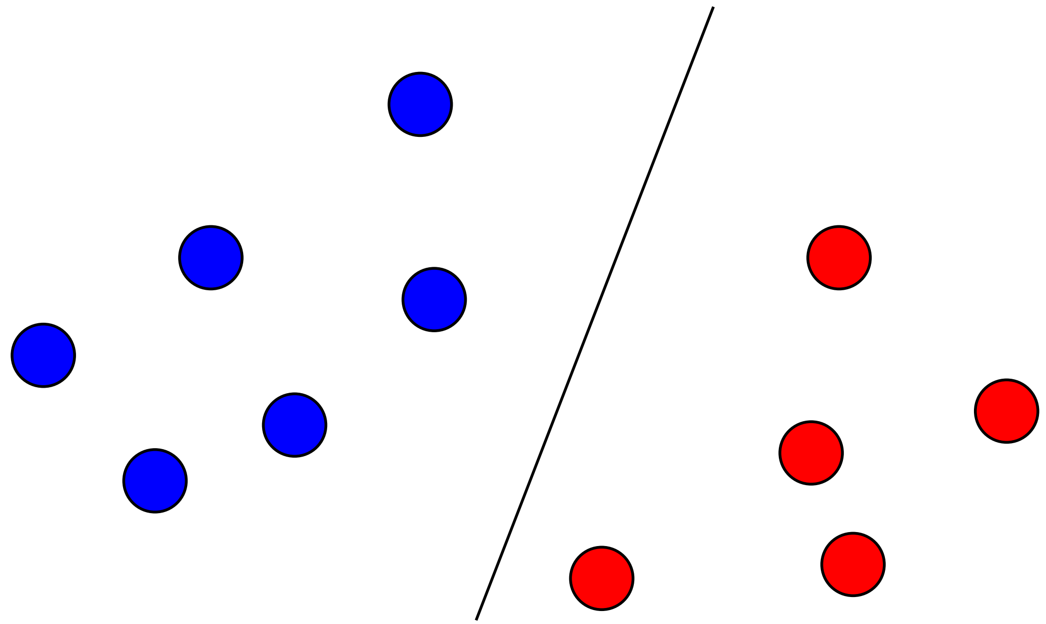
Linear classifier, some degrees of freedom



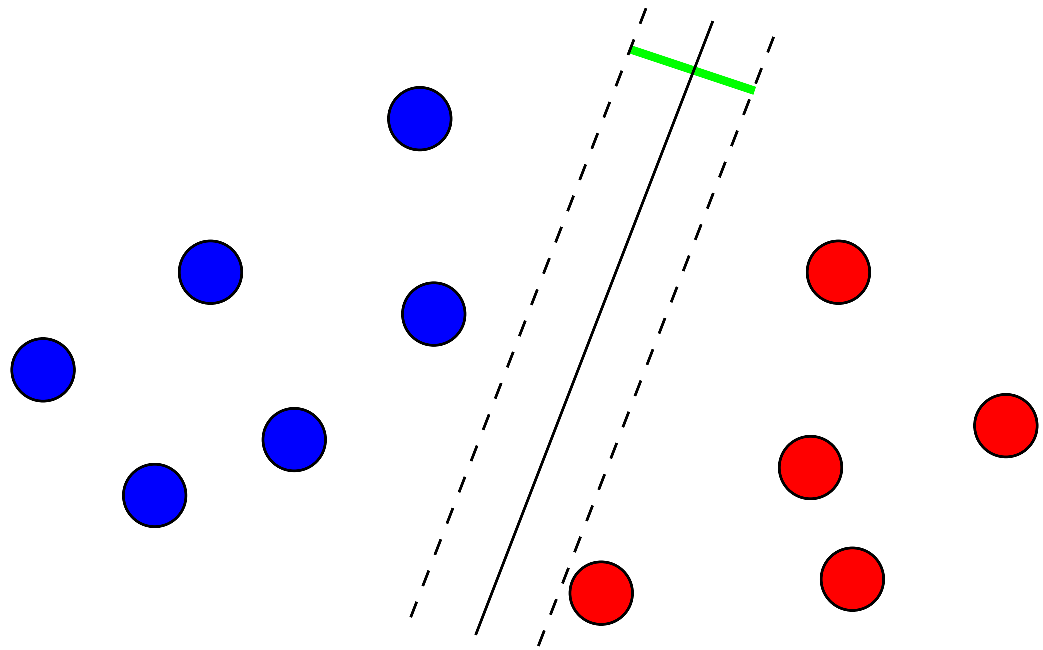
Which one is better?



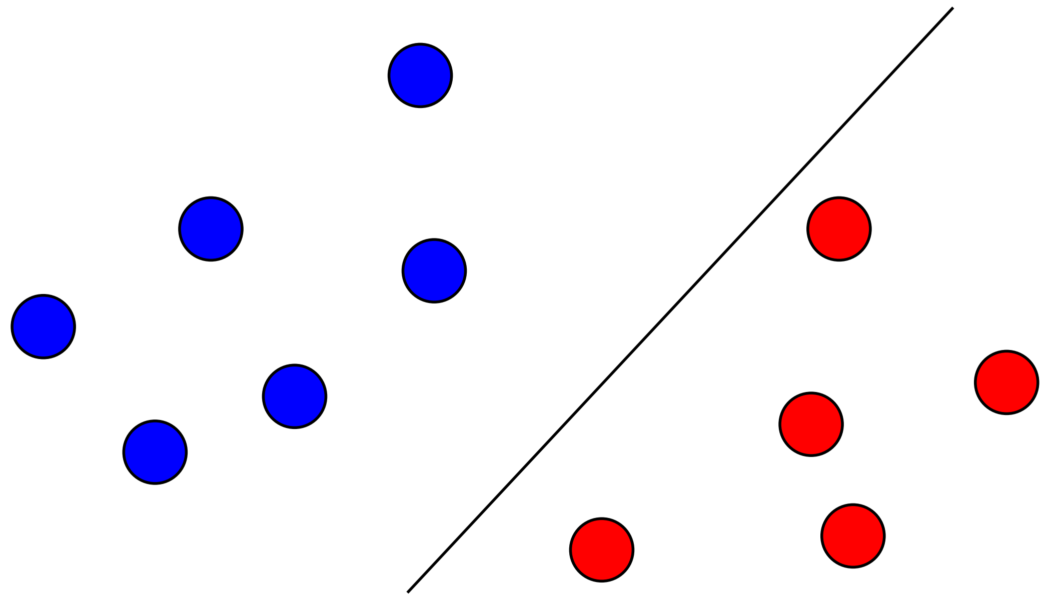
A criterion to select a linear classifier: the margin



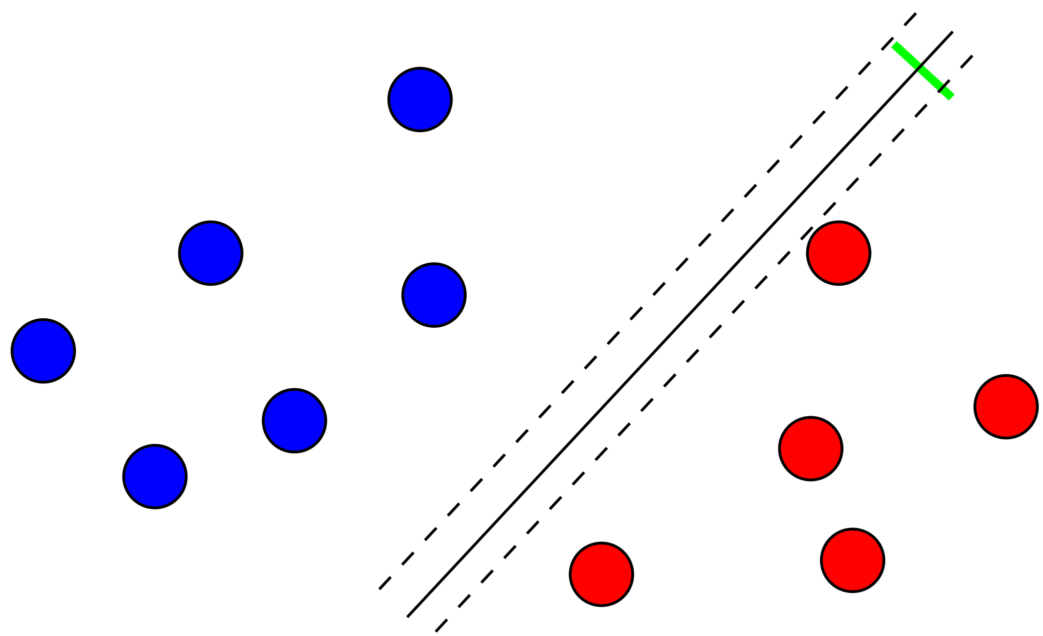
A criterion to select a linear classifier: the margin



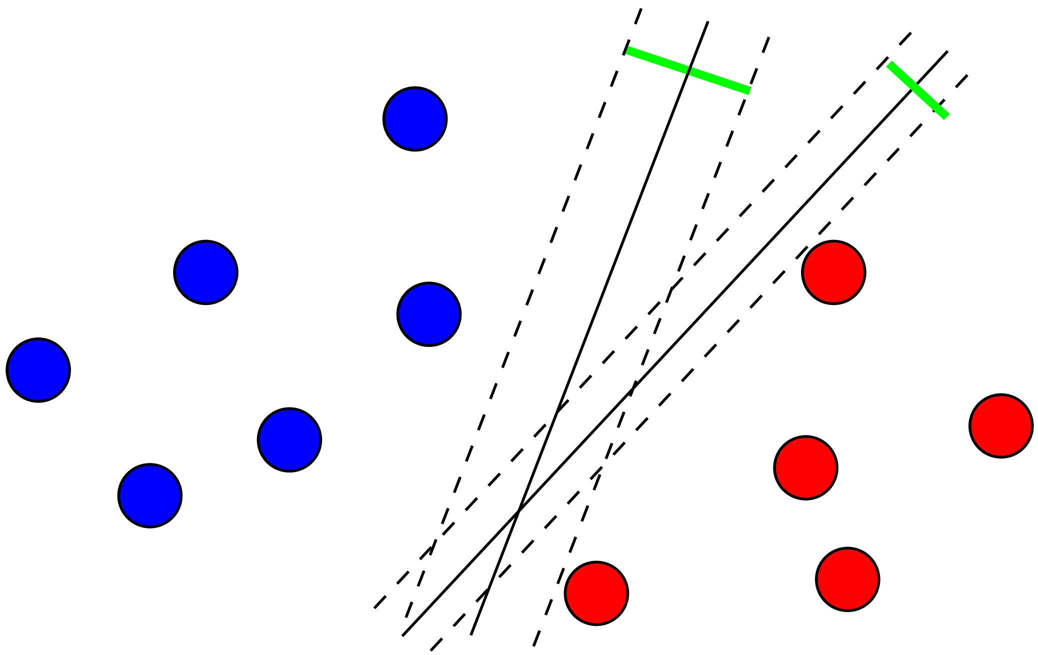
A criterion to select a linear classifier: the margin



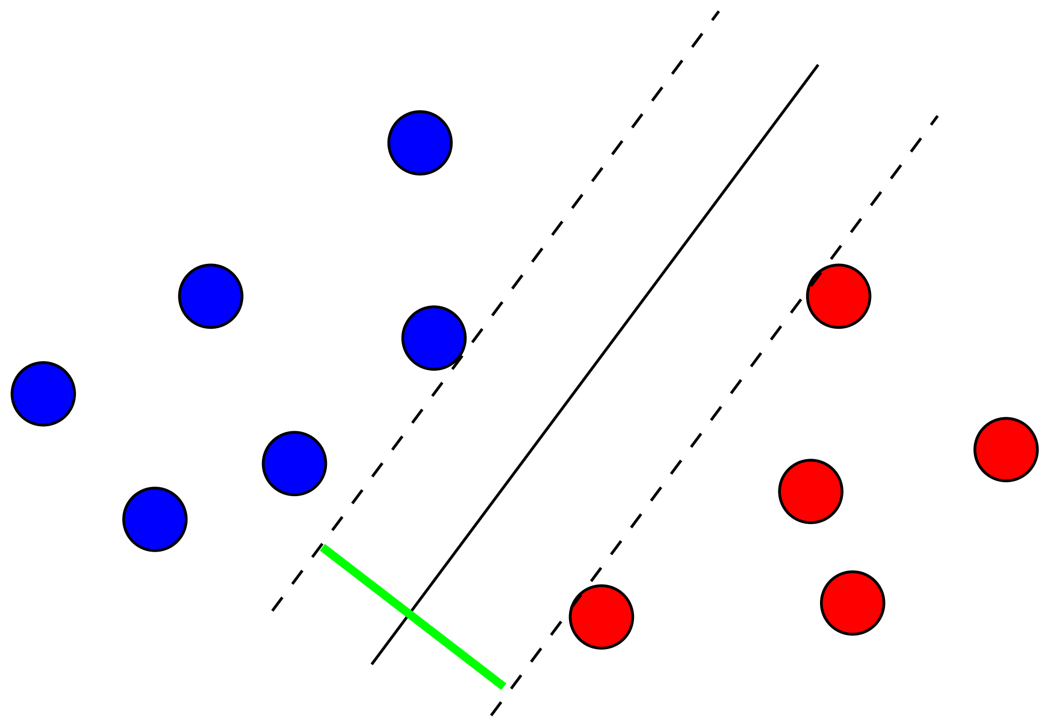
A criterion to select a linear classifier: the margin



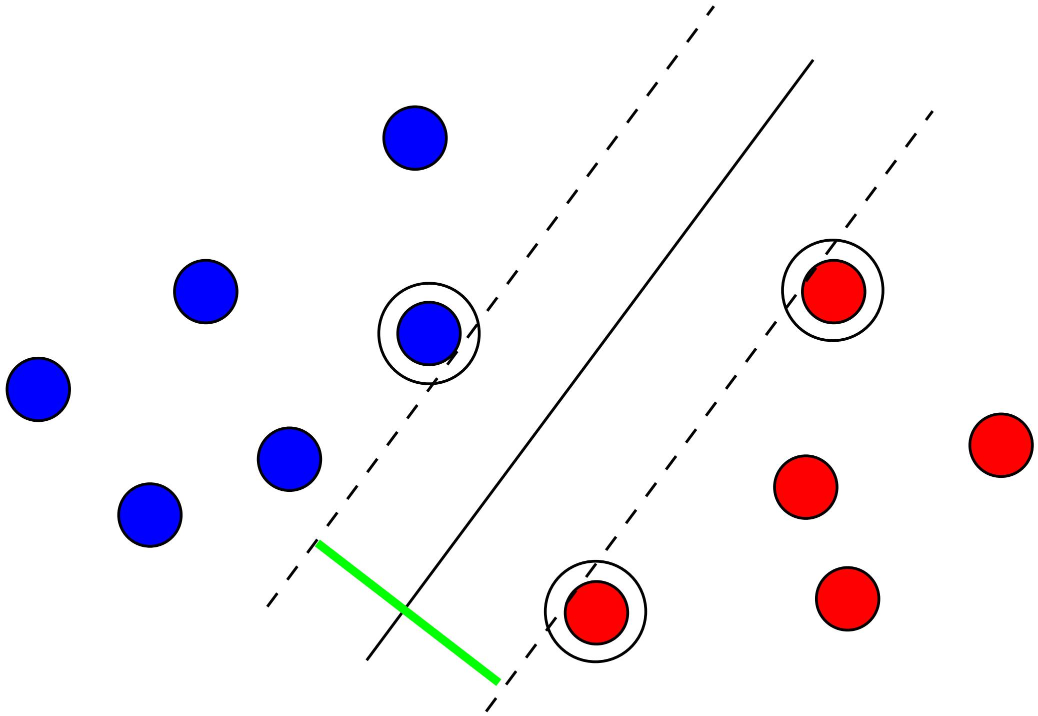
A criterion to select a linear classifier: the margin



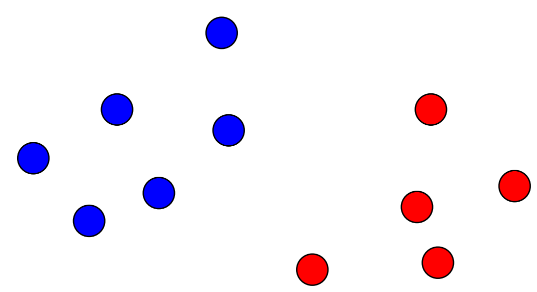
Largest Margin Linear Classifier



Support Vectors with Large Margin



In equations



- The **training set** is a finite set of n data/class pairs:

$$\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\},$$

  where $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{y}_i \in \{-1, 1\}$.

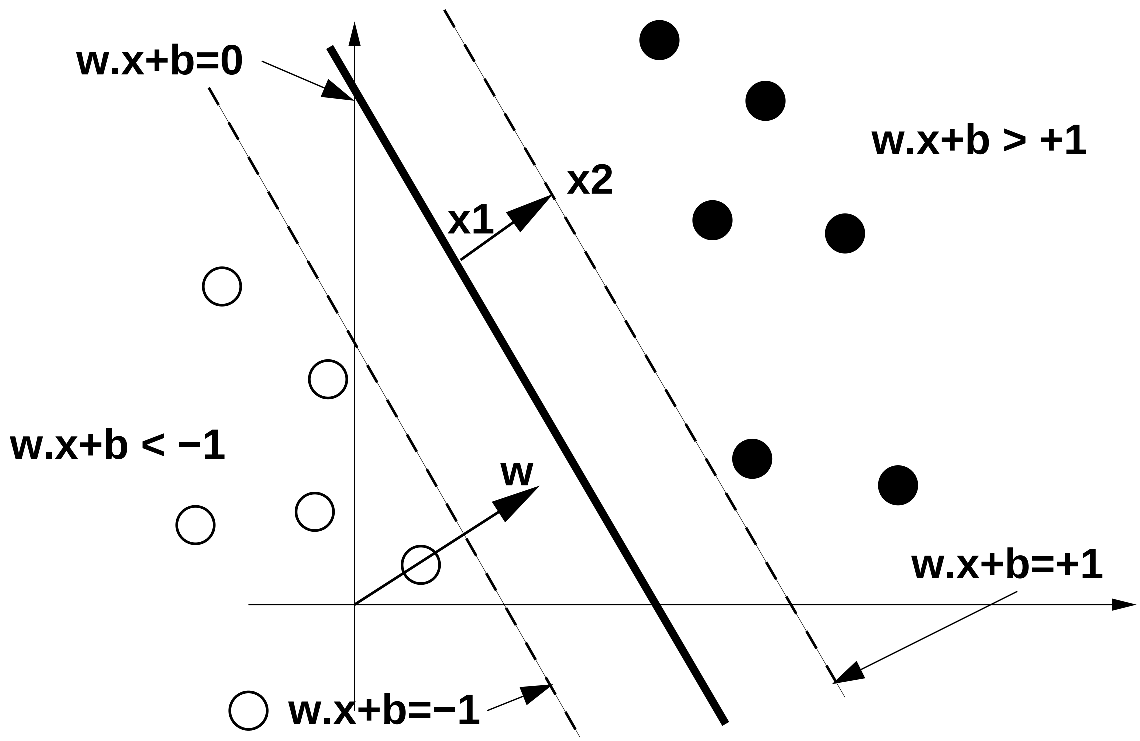
- We assume (for the moment) that the data are **linearly separable**, i.e., that there exists $(\mathbf{w}, b) \in \mathbb{R}^d \times \mathbb{R}$ such that:

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b > 0 & \text{if } \mathbf{y}_i = 1, \\ \mathbf{w}^T \mathbf{x}_i + b < 0 & \text{if } \mathbf{y}_i = -1. \end{cases}$$

How to find the largest separating hyperplane?

For the linear classifier $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ consider the *interstice* defined by the hyperplanes

- $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = +1$
- $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = -1$



The margin is $2/||\mathbf{w}||$

- Indeed, the points $\mathbf{x}_1$ and $\mathbf{x}_2$ satisfy:

$$\begin{cases} \mathbf{w}^T\mathbf{x}_1 + b = 0, \\ \mathbf{w}^T\mathbf{x}_2 + b = 1. \end{cases}$$

- By subtracting we get $\mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) = 1$, and therefore:

$$\gamma = 2||\mathbf{x}_2 - \mathbf{x}_1|| = \frac{2}{||\mathbf{w}||}.$$

where γ is the margin.

All training points should be on the appropriate side

- For positive examples ($y_i = 1$) this means:

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1$$

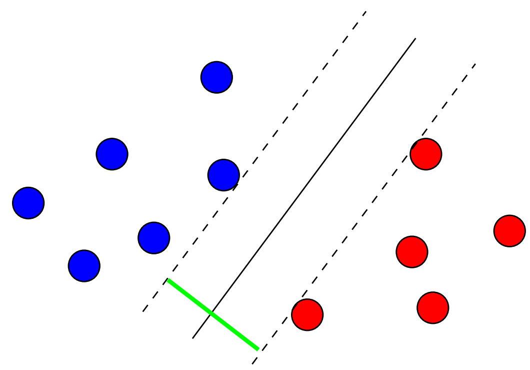
- For negative examples ($y_i = -1$) this means:

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1$$

- in both cases:

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

Finding the optimal hyperplane



- Find $(\mathbf{w}, b)$ 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:

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

- introduce **one dual variable α_i for each constraint**,
- namely, for **each training point**. The Lagrangian is, for $\alpha \succeq 0$,

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

The Lagrange dual function

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

is only defined when

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

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

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

- To compute the dual, just maximize g w.r.t. α.
- Strong duality holds. KKT gives us $\alpha_i(\mathbf{y}_i \mathbf{w}^T \mathbf{x}_i - 1) = 0$, either $\alpha_i = 0$ or $\mathbf{y}_i \mathbf{w}^T \mathbf{x}_i = 1$.
- $\alpha_i \neq 0$ **only for points on the support hyperplanes** $\{(\mathbf{x}, \mathbf{y}) | \, \mathbf{y}\mathbf{w}^T \mathbf{x}_i = 1\}$.

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 on $\mathbb{R}^n$, with *box constraints*.
α^* can be found efficiently using dedicated optimization softwares

Recovering the optimal hyperplane

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

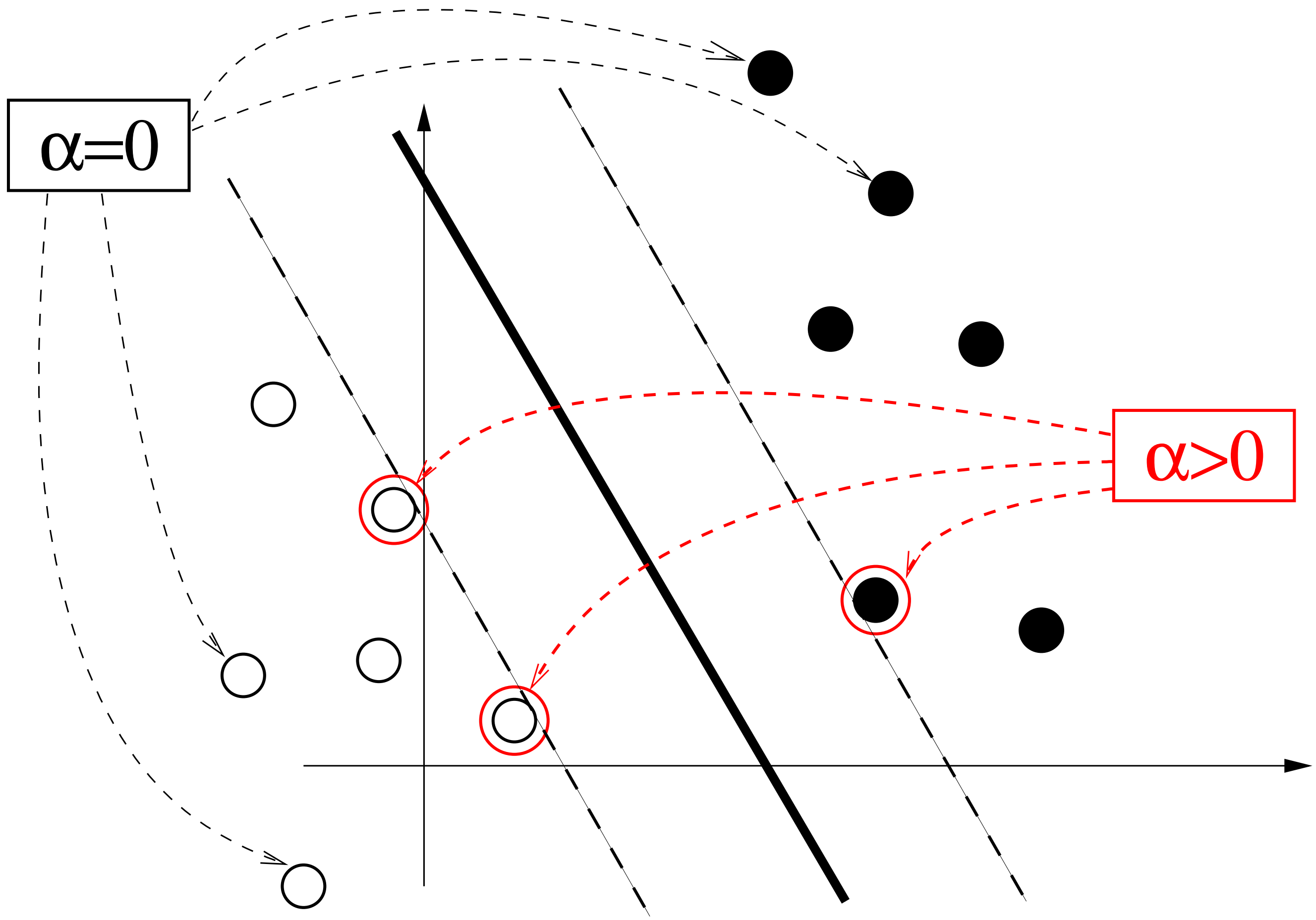
$$b^* = -\frac{1}{2}\left(\min_{\mathbf{y}_i=1,\alpha_i>0}(\mathbf{w}^T\mathbf{x}_i) + \max_{\mathbf{y}_i=-1,\alpha_i>0}(\mathbf{w}^T\mathbf{x}_i)\right)$$

- the **decision function** is therefore:

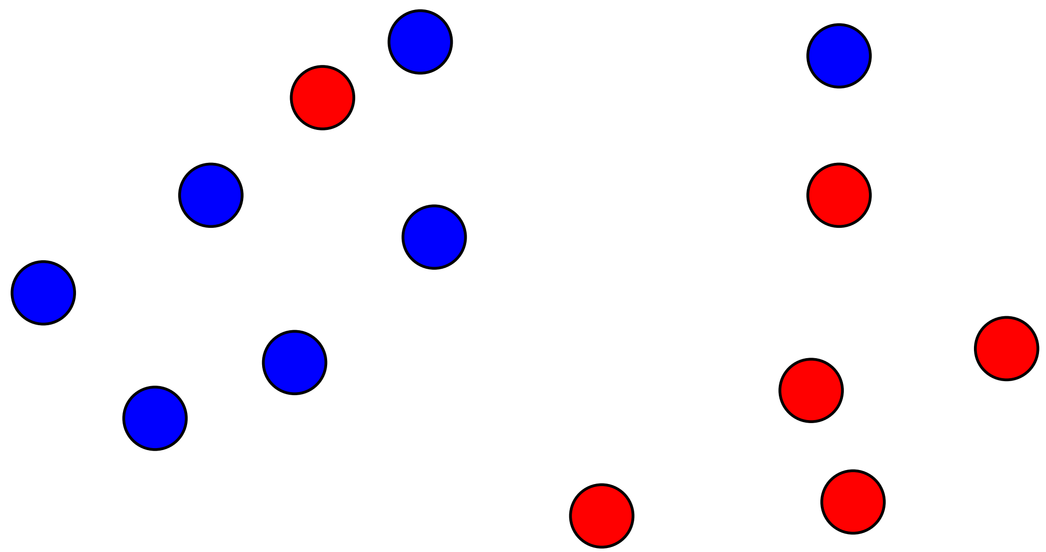
$$\begin{aligned} f^*(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} + b^* \\ &= \sum_{i=1}^n \alpha_i \mathbf{x}_i^T\mathbf{x} + b^*. \end{aligned}$$

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

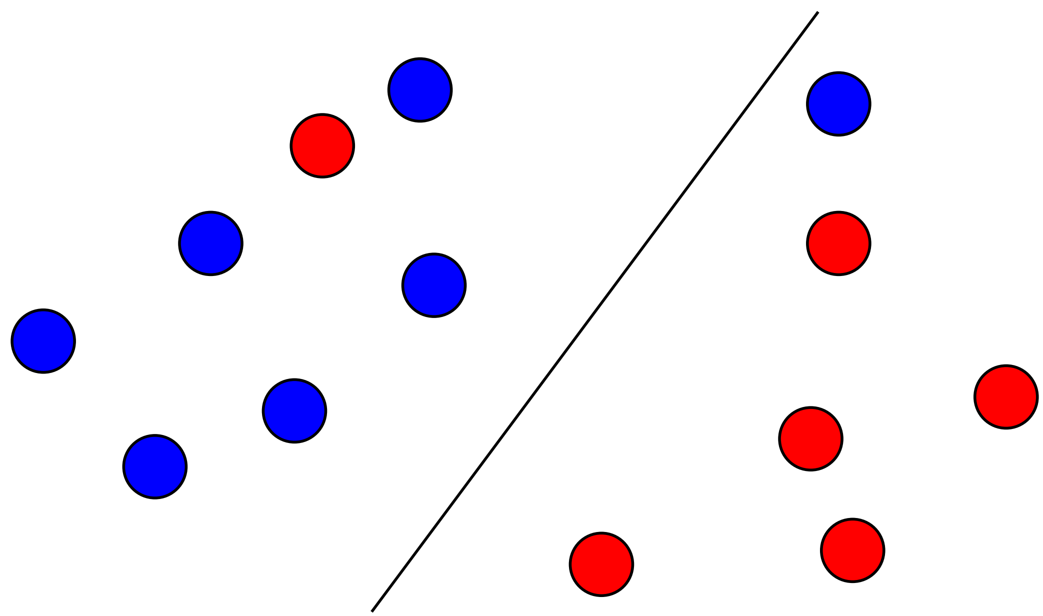
Interpretation: support vectors



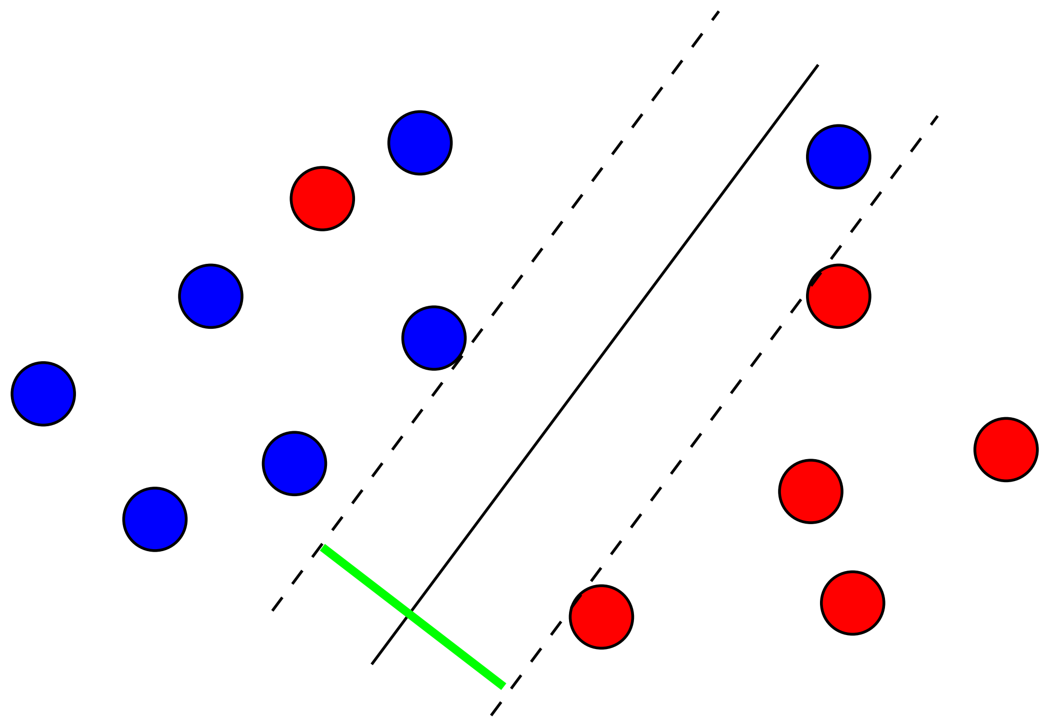
What happens when the data is not linearly separable?



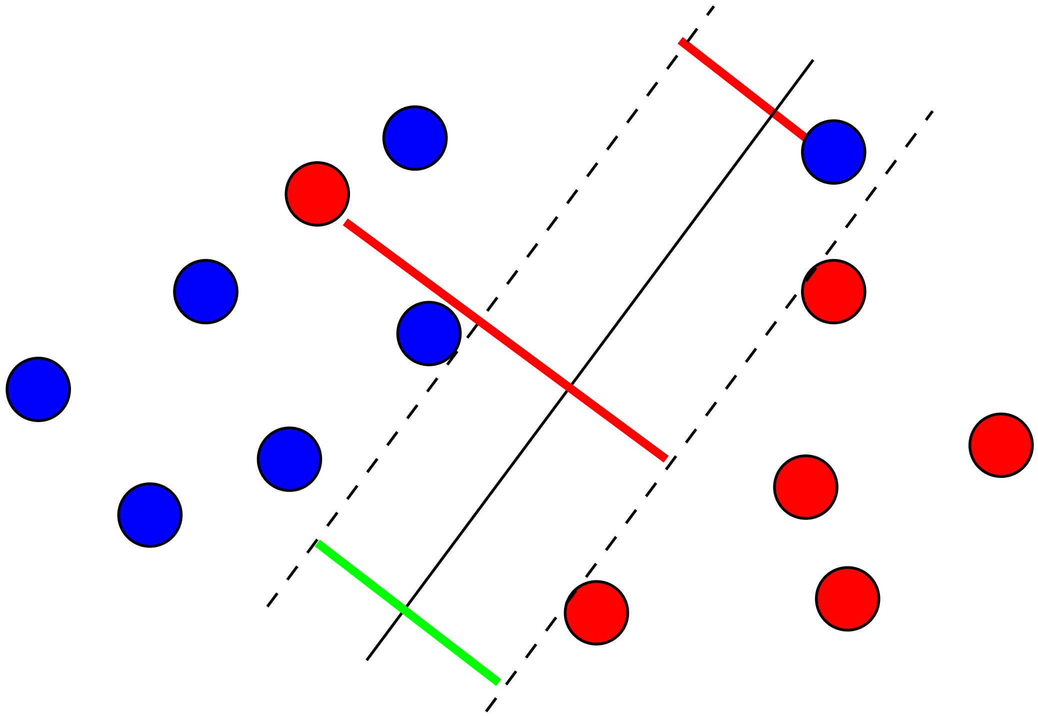
What happens when the data is not linearly separable?



What happens when the data is not linearly separable?

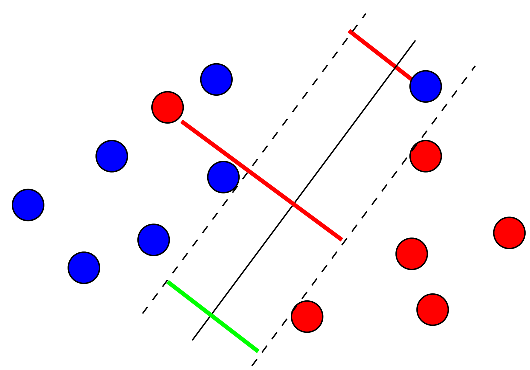


What happens when the data is not linearly separable?



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

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

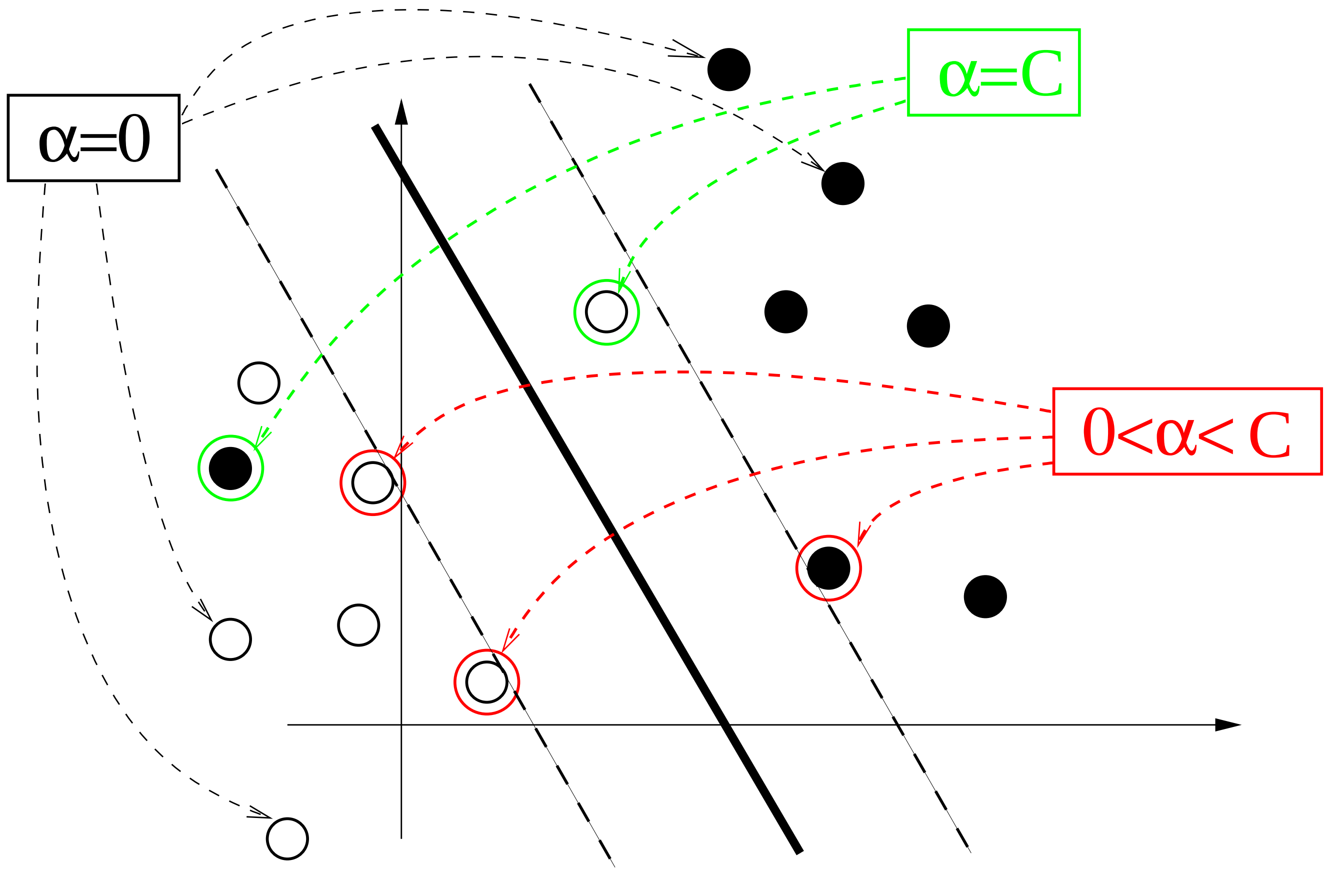
- In that case the dual function

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

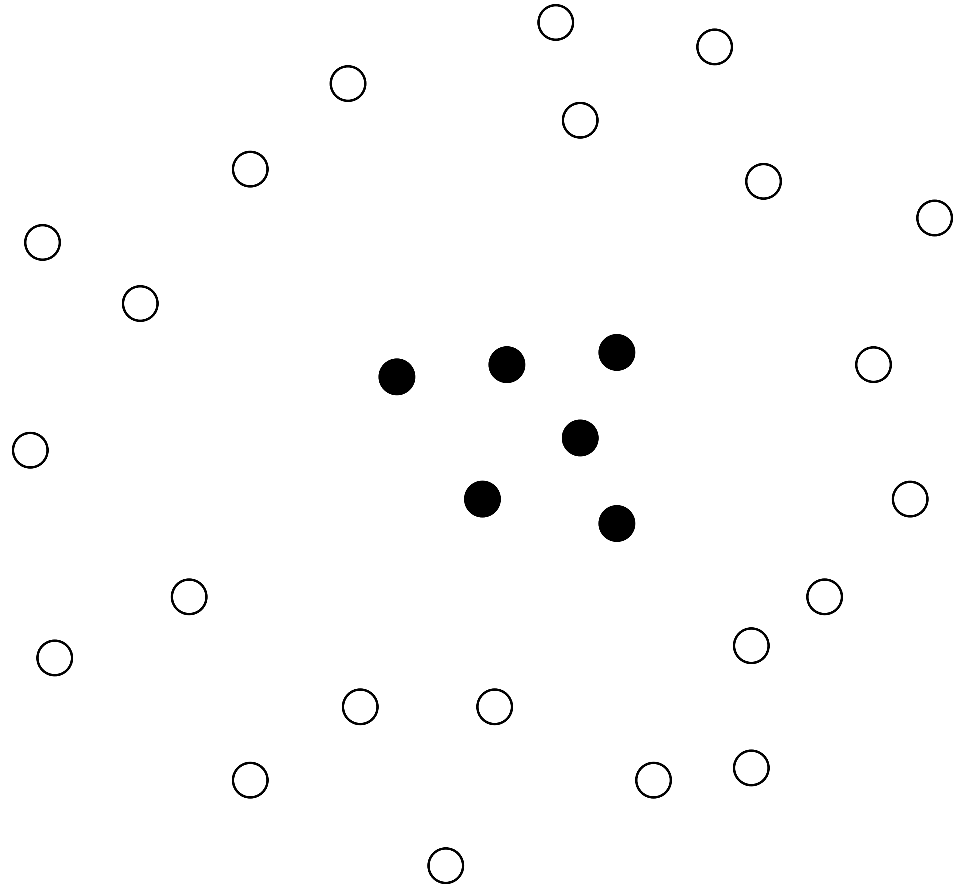
which is finite under the constraints:

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

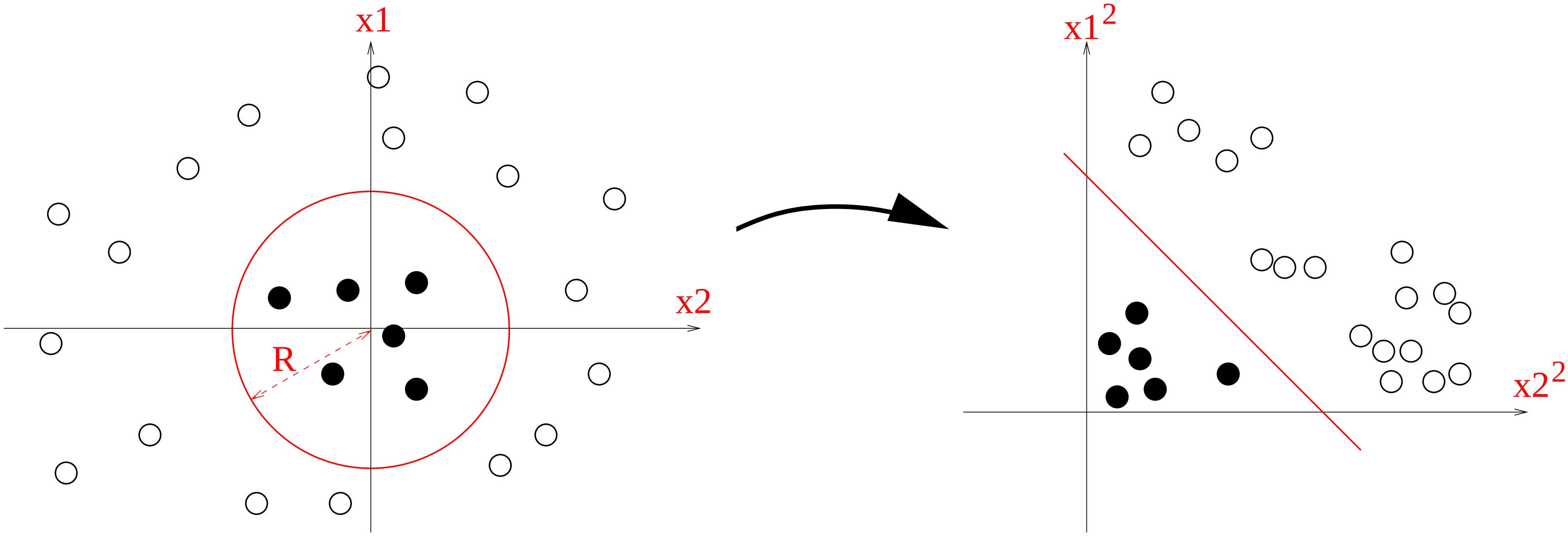
Interpretation: bounded and unbounded support vectors



Sometimes linear classifiers are not interesting



Solution: non-linear mapping to a feature space



Let $\phi(\mathbf{x}) = (x_1^2, x_2^2)'$, $\mathbf{w} = (1,1)'$ and $b = 1$. Then the decision function is:

$$f(\mathbf{x}) = x_1^2 + x_2^2 - R^2 = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b,$$

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 $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}')$

- 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:

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

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 K and $\mathbf{y}$:

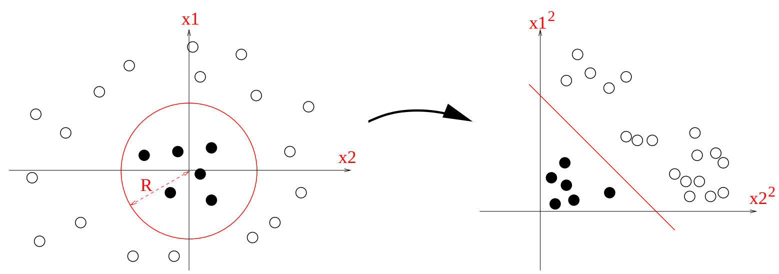
$$\begin{array}{ll} \text{maximize} & g(\alpha) = \alpha^T \mathbf{1} - \frac{1}{2}\alpha^T(\mathbf{y}^T \boldsymbol{K} \mathbf{y})\alpha \\ \text{such that} & 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i \mathbf{y}_i = 0. \end{array}$$

- in the end the solution $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \cdot) + b$.

Kernel example: polynomial kernel

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

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



Empirical Risk Minimization

- Starting with $\{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$, n couples of $\mathcal{X} \times \mathcal{Y}$,
- A functional class $\mathcal{F}$,
- A cost function $\boldsymbol{c} : \mathcal{Y} \times \mathcal{Y}$, $c \geq 0$, which penalizes discrepancies (distances? squared-distance?)
- find the function which minimizes

$$\hat{f} \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{c}(f(\mathbf{x}_i), y_i)$$

  and use this f as a decision function.

- As usual in minimizations, we love:
  - Convex problems, unique minimizers
  - Stable solutions numerically.

Linear least squares

- When $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$,
- $\boldsymbol{\mathcal{F} = \{\mathbf{x} \mapsto \beta^T \mathbf{x} + b\,, \beta \in \mathbb{R}^d, b \in \mathbb{R}\}}$, $\boldsymbol{c(\mathbf{y}_1, \mathbf{y}_2) = \|\mathbf{y}_1 - \mathbf{y}_2\|^2}$,
- The problem is known as **regression** with the **least squares criterion**.
- In this case, the minimizer

$$\underset{f \in \mathcal{F}}{\operatorname{argmin}} \hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{c}(f(\mathbf{x}_i), \mathbf{y}_i)$$

is **unique** (assuming $n > d$), and is equal to

$$\begin{bmatrix} b \\ \beta \end{bmatrix} = (XX^T)^{-1} X \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{bmatrix}$$

where $X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}$

Minimizers on general functional classes

- In this case a few factors contribute to the uniqueness:
  - convexity of c,
  - the feasible set, $\mathcal{F}$ is sufficiently **small** to show no-degeneracy.
- Imagine we use instead a RKHS for $\mathcal{F}$.
- Usually two sources of problems:
  - selecting functions in (infinite dimensional) RKHS can be ill-posed:

$$\mathrm{card}\{\underset{f\in\mathcal{H}}{\operatorname{argmin}}\,\hat{R}(f)\} \text{ could be } \infty$$

  - within these solutions, some are more desirable than others. In particular, better select smoother functions.

Minimizers in RKHS

- Main message: we do not want to deal with problems of optimization in **infinite dimensional** Hilbert spaces using **finite numbers of constraints**.
- Two major intuitions:

$$\boxed{\text{Bias the selection towards functions of low norm } \|f\|_{\mathcal{H}}}$$

- the norm quantifies the **roughness** of the function.
- if possible, better choose a **smooth** function for a decision function.

Minimizers in RKHS

$\boxed{\text{Bias the selection towards functions we know in } \mathcal{H}\text{, namely } \mathcal{H}_n}$

- When the criterion only depends on the values of f on a sample $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}$, as in $\hat{R}$, under certain conditions,

$$\operatorname{argmin} \hat{R} \subset \mathcal{H}_n \stackrel{\text{def}}{=} \operatorname{span}\{k(\mathbf{x}_i, \cdot)_{i=1,\cdots,n}\}.$$

- As a consequence, f can be selected within the optimum set

$$\underset{f \in \mathcal{H}_{\boldsymbol{n}}}{\operatorname{argmin}} \hat{R}(f),$$

$\mathcal{H}_{\boldsymbol{n}}$ is a **finite** dimensional subspace of $\mathcal{H}$. Always easier to handle mathematically.

Representer Theorem

Theorem 1. *Let* $\{x_i\}_{1\leq i\leq n}$ *be points in* $\mathcal{X}$ *and let* $\Psi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ *be any function that is strictly increasing with respect to its last argument. Then any solution to the problem*

$$\min_{f\in\mathcal{H}} \Psi\left(f(x_1), \cdots, f(x_n), \|f\|_{\mathcal{H}_k}\right)$$

is in $\mathcal{H}_n$.

Proof. Let $f = f_n + f^\perp$, where $f_n \in \mathcal{H}_n, f^\perp \in \mathcal{H}_n^\perp$.

- We have that $f(x_i) = f_n(x_i)$ since

$$f(x_i) = \langle f, k(x_i, \cdot)\,\rangle = \langle f, k(x_i, \cdot)\,\rangle = \langle f_n, k(x_i, \cdot)\,\rangle + \langle f^\perp, k(x_i, \cdot)\,\rangle = f_n(x_i).$$

Hence for any function $f \in \mathcal{H}$, $\Psi(f_n) < \Psi(f)$ hence any optimal $f^\star$ must be such that $f^\star \in \mathcal{H}_n$.

Empirical Risk Minimization

- We can now write for a strictly convex loss $\boldsymbol{c}$,

$$\hat{f} = \underset{f \in \boldsymbol{\mathcal{H}_n}}{\operatorname{argmin}} \hat{R}_\lambda(f) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{c}(f(\mathbf{x}_i), y_i) + \lambda \|f\|_{\mathcal{H}}^2$$

  and this $\hat{f}$ is **unique**

- $\lambda > 0$ balances the tradeoff between
  - a good fit for the data at hand
  - a smoothness as measured by $\|f\|$.

- This formulation can be generalized to any measure of smoothness J on $\mathcal{F}$,

$$R_c^\lambda(f) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} c\left(f(x_i), y_i\right) + \lambda J(f).$$

A few examples

- $\mathcal{X}$ is Euclidian, $\mathcal{Y} = \mathbb{R}$, $\mathcal{F} = \mathcal{X}^*$, the dual of $\mathcal{X}$ and $c(f(x), y) = (y - f(x))^2$, minimizing R_c^λ is known as
    - least-square regression when $\lambda = 0$;
    - ridge regression when $\lambda > 0$ and J is the Euclidian 2-norm;
    - the lasso when $\lambda > 0$ and J is the 1-norm.

- $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \mathbb{R}$, $\mathcal{F}$ is the space of m-times differentiable functions on $[0, 1]$ and $J = \int_{[0,1]} \left(f^{(m)}(t)\right)^2 dt$, we obtain regression by natural splines of order m.

A few examples

- $\mathcal{X}$ is a set endowed with a kernel k and $\mathcal{Y} = \{-1, 1\}$, $\mathcal{F} = \mathcal{H}$, $J = \|\cdot\|_{\mathcal{H}}$ and
  - the hinge loss $c(f(x), y) = (1 - yf(x))^+ \rightarrow$ SVM
  - $c(f(x), y) = (y - f(x))^2 \rightarrow$ LS-SVM,
  - $c(f(x), y) = \ln(1 + e^{-yf(x)}) \rightarrow\rightarrow$ kernel logistic regression.

- When $\mathcal{X}$ is an arbitrary set endowed with a kernel k and $\mathcal{Y} = \mathbb{R}$, $\mathcal{F} = \mathcal{H}$, $J = \|\cdot\|_{\mathcal{H}}$ and $c(f(x), y) = (|y - f(x)| - \varepsilon)^+$, the ε-insensitive loss function, the solution to this program is known as support vector regression.

Principal Component Analysis in $\mathbb{R}^d$.

- Start from a sample $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$.
- Look for directions $v_1, \cdots, v_d$ of $\mathbb{R}^d$ such that for $1 \leq j \leq d$,

$$v_j = \underset{v \in \mathbb{R}^d, \|v\|=1, v \perp \{v_1, \cdots, v_{j-1}\}}{\operatorname{argmax}} \underset{X}{\mathbf{var}}[v^T \mathbf{x}],$$

- For $f : \mathbb{R}^d \to \mathbb{R}$, $\quad \mathbf{var}_X[f]$ is the **empirical variance** w.r.t. sample X, that is

$$\underset{X}{\mathbf{var}}[f] = E_X(f(\mathbf{x}) - E_X[f(\mathbf{x})])^2 = \frac{1}{n}\sum_{i=1}^{n}\left(f(\mathbf{x}_i) - \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)\right)^2].$$

- The r first eigenvectors $v_1, \cdots, v_r$ are the principal components.

Canonical Correlation Analysis in $\mathbb{R}^{d,d'}$.

- **Two associated samples** X paired with $Y = \{\mathbf{y}_i\}_{1\leq i\leq n}$ in $\mathbb{R}^{d'}$,
- Assume that the pairs (x_i, y_i) are drawn from a i.i.d law.
- CCA looks for relationships between X and Y by looking for linear projections of the samples X and Y,

$$\alpha^T \mathbf{x}_i \text{ and } \beta^T \mathbf{y}_j,$$

such that $\mathbf{corr}(\alpha^T \mathbf{x}_i, \beta^T \mathbf{y}_i)$ is **high**.

$$\begin{aligned}(\alpha, \beta) &= \underset{\xi\in\mathbb{R}^d,\zeta\in\mathbb{R}^{d'}}{\operatorname{argmax}} \ \mathbf{corr}_{X,Y}[\alpha^T, \beta^T] \\ &= \underset{\xi\in\mathbb{R}^d,\zeta\in\mathbb{R}^{d'}}{\operatorname{argmax}} \frac{\mathbf{cov}_{X,Y}[\alpha^T, \beta^T]}{\sqrt{\mathbf{var}_X[\alpha^T]\,\mathbf{var}_Y[\beta^T]}}\end{aligned}$$

where for two real valued functions $f : \mathcal{X} \rightarrow \mathbb{R}$ and $g : \mathcal{Y} \rightarrow \mathbb{R}$ we write

$$\underset{X}{\mathbf{var}}[f] = E_X(f(\mathbf{x}) - E_X[f(\mathbf{x})])^2 = \frac{1}{n}\sum_{i=1}^{n}\left(f(x_i) - \frac{1}{n}\sum_{j=1}^{n} f(\mathbf{x}_j)\right)^2,$$

$$\underset{Y}{\mathbf{var}}[g] = E_X(g(\mathbf{y}) - E_Y[g(\mathbf{y})])^2 = \frac{1}{n}\sum_{i=1}^{n}\left(g(y_i) - \frac{1}{n}\sum_{j=1}^{n} g(\mathbf{y}_j)\right)^2,$$

$$\begin{aligned}\underset{X,Y}{\mathbf{cov}}[f,g] &= E_{X,Y}[(f(\mathbf{x}) - E_X[f(\mathbf{x})])(g(\mathbf{y}) - E_Y[g(\mathbf{y})])] \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(f(x_i) - \frac{1}{n}\sum_{j=1}^{n} f(\mathbf{x}_j)\right)\left(g(y_i) - \frac{1}{n}\sum_{j=1}^{n} g(\mathbf{y}_j)\right)\end{aligned}$$

Unsupervised Techniques

both **non-convex** optimizations look for **vectors** in $\mathbb{R}^d$, that is **linear projections** which summarize the data.

- Although non-convex, the optima can be computed through eigenvalue decompositions of matrices.
- Courant-Weyl-Fisher minimax principle for Rayleigh quotients.

- Yet, these tools have limitations: linearity.

Kernel methods allow us to study nonlinear eigenfunctions and CCA-projections

kernel- Principal Component Analysis [SSM98]

- Consider X as spanning $\mathcal{H}_n$ the two previous optimizations become

$$f_j = \operatorname*{argmax}_{f\in\mathcal{H}_{\mathcal{X}}, \|f\|_{\mathcal{H}_{\mathcal{X}}}=1, f\perp\{f_1,\cdots,f_{j-1}\}} \operatorname*{\mathbf{var}}_X[\langle f, k_{\mathcal{X}}(\mathbf{x},\cdot)\rangle_{\mathcal{H}}],$$

  for $1\leq j\leq n$.

- Using the $n\times n$ kernel matrix K_X, more precisely its centered counterpart

$$\bar{K}_X = (I_n - \frac{1}{n}\mathbb{1}_{n,n})K_X(I_n - \frac{1}{n}\mathbb{1}_{n,n}).$$

  The eigenfunctions f_i are recovered through the eigenvalue/eigenvector pairs (e_i, d_i) of $\bar{K}_X$,

$$\bar{K}_X = EDE^T$$

  where $D = \mathbf{diag}(d)$ and E is an orthogonal matrix. Writing $U = ED^{-1/2}$ we have that

$$f_j(\cdot) = \sum_{i=1}^{n} U_{i,j}k(x_i,\cdot)$$

  with $\mathbf{var}_X[f_j(x)] = \frac{d_j}{n}$.

kernel- Canonical Correlation Analysis [Aka01,BJ02]

- A direct adaptation of the CCA criterion to infinite dimensional RKHS,

$$(f,g) = \underset{f\in\mathcal{H}_\mathcal{X}, g\in\mathcal{H}_\mathcal{Y}}{\operatorname{argmax}} \frac{\mathbf{cov}_{X,Y}[\langle f, k_\mathcal{X}(x,\cdot)\rangle_{\mathcal{H}_\mathcal{X}}, \langle g, k_\mathcal{Y}(y,\cdot)\rangle_{\mathcal{H}_\mathcal{X}}]}{\sqrt{\mathbf{var}_X[\langle f, k_\mathcal{X}(x,\cdot)\rangle_{\mathcal{H}_\mathcal{X}}]\,\mathbf{var}_Y[\langle g, k_\mathcal{Y}(y,\cdot)\rangle_{\mathcal{H}_\mathcal{Y}}]}}.$$

- This does not work numerically on finite samples. Denominator goes to zero.
- In [FBG07], it is shown that using

$$(f,g) = \underset{f\in\mathcal{X}, g\in\mathcal{Y}}{\operatorname{argmax}} \frac{\mathbf{corr}_{X,Y}[f,g]}{\sqrt{(\mathbf{var}_X[f] + \lambda\|f\|^2)(\mathbf{var}_Y[g] + \lambda\|g\|^2)}},$$

and letting $\lambda \to 0$ as $n \to \infty$ works.

kernel-Canonical Correlation Analysis [Aka01,BJ02]

- The finite sample estimates f^n and g^n can be recovered as

$$f^n(\cdot) = \sum_{i=1}^{n} \xi_i \varphi_i(\cdot),$$

$$g^n(\cdot) = \sum_{i=1}^{n} \zeta_i \psi_i(\cdot)$$

where ξ and ζ are the solutions of

$$(\xi, \zeta) = \underset{\substack{\xi,\zeta \in \mathbb{R}^n, \\ \xi^T(\bar{K}_X^2 + n\lambda \bar{K}_X)\xi = \zeta^T(\bar{K}_Y^2 + n\lambda \bar{K}_Y)\zeta = 1}}{\operatorname{argmax}} \zeta^T \bar{K}_Y \bar{K}_X \xi$$

and

$$\varphi_i(\cdot) = k_{\mathcal{X}}(\mathbf{x}_i, \cdot) - \frac{1}{n}\sum_{j=1}^{n} k_{\mathcal{X}}(\mathbf{x}_i, \cdot), \quad \psi_i(\cdot) = k_{\mathcal{Y}}(\mathbf{y}_i, \cdot) - \frac{1}{n}\sum_{j=1}^{n} k_{\mathcal{Y}}(\mathbf{y}_i, \cdot),$$

are the centered projections of $(\mathbf{x}_i)$ and $(\mathbf{y}_j)$ in $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$