

Language Information Processing, Advanced


Text Classifiers


mcuturi@i.kyoto-u.ac.jp


Today's talk

- Objective: supervised inference on text data.
 - **Ex.1** Given a large database of **news articles** about *business, sports, literature, politics etc.etc.*
 - ▷ Build a system that can classify **automatically** new articles.

Business »


 **AP reports pipeline operations can restart**
KHAS-TV - 8 hours ago
The Associated Press reported today that regulators allowed a keystone oil pipeline to restart operations. This co regulators blocked TransCanada-which owns the line-from restarting until repairs were made.
CTV.ca

 **GMail hacking draws FBI interest**
Economic Times - 19 hours ago
WASHINGTON: The computer phishing scam that Google says originated in China was directed at an unknown n staff officials and set off the FBI inquiry that began this week, according to several administration officials.
Globe and ...

 **Giant open-pit mine raises questions in Uruguay**
AFP - 10 hours ago
CERRO CHATO, Uruguay - A plan to build a giant open pit mine has created a sharp rift between those who think agricultural land should be protected, and those wanting to exploit its wealth. The Aratiri project, owned by Zamin
AFP

[More Business stories](#)

Sci/Tech »

 **WWDC, iPhone 5 in limelight: What new Android smartphones are lined up?**
International Business Times - 2 hours ago
By IB Times Staff Reporter | June 5, 2011 7:02 AM EDT All eyes are on Apple Worldwide Developer Conference on Monday and whether Steve Jobs will unveil Apple iPhone 5. Most observers say Apple will not unveil the next
Information...

Today's talk

- **Ex.2** Given a large set of e-mails in a mailbox, *family, friends, spam, ads, newsletters etc.*
 - ▷ Build a system that **categorizes** automatically a new email.

Quality Medicine Available C7 - The most complete Pha
Viagra Professional as low as \$3.84 - Visit our new or
制-造-型-企-业-车-间-管-理-技-能-高-级-训-练-GB2312?B
Re[13]: - Hot selling meds at cheap All countries shipping
Our store is your cureall! - My Canadian Pharmacy We
We offer a variety of different licenses and discounts
Effortless Discount Offerings xh - Check Out our new
(no subject) -
We will help you get laid - Hey there. Just came across t
Lively Benefits of Creativity - When it comes to corpora
FW:hope you didn't mind - usasia , lcmd the lisi or ype b
RE: What's new out there? - tv-channel may dokeytomp
Looking to ReFi or a Home Equity Loan? - isn't some l
We cure any desease! - My Canadian Pharmacy We shi
Unlimited Systemworks Downloads, get your 70% di
cheap oem soft shipping //orldwide - TOP 10 NEW TIT

Today's talk

- **Ex.3** Given a set of requests/messages sent to a retailer: complaints, need for technical support, praise
 - ▷ Build a system that **forwards directly** the message to the relevant department.
- Who is interested in this? internet companies, companies with large customer support, polling institutions *etc.*

Approaches presented today:
naive bayes, large margin classifiers, extensions with kernels

Text classification & probabilistic framework

- Assume that there is a probability p_{text} on texts on the internet

Today will be a rainy day

In Ecuador tiger-hunters enjoy eating marshmallows

Buffalo buffalo Buffalo buffalo buffalo Buffalo buffalo Buffalo buffalo

- A probability quantifies how **likely** sentences are to appear
- Any idea on how this likelihood might be measured?

Text classification & probabilistic framework

- This probability takes into account **grammar and meaning**.
- Search engines are useful to have an idea about p_{text}

Today will be a rainy day

"today will be a rainy day"

About 288,000 results (0.24 seconds)

In Ecuador tiger-hunters enjoy eating marshmallows

"In Ecuador tiger-hunters enjoy eating marshmallows"

▶ ⚠ No results found for "In Ecuador tiger-hunters enjoy eating marshmallows".

Buffalo buffalo Buffalo buffalo buffalo Buffalo buffalo Buffalo buffalo

"Buffalo buffalo Buffalo buffalo buffalo Buffalo buffalo Buffalo buffalo"

About 4,980,000 results (0.29 seconds)

Text classification & probabilistic framework

- We assume that there is **something to learn from data** (supervised inference)
- We assume our task is to categorize a given text among C given classes
 - agriculture, computer chips, energy, environment, sports, politics, gossip *etc.*
 - friends, family, spam, advertisements, newsletters *etc.*

- We also assume there is a probability p_{cat} **on categories.**

Text classification & probabilistic framework

- We assume that there is **something to learn from data** (supervised inference)
- We assume our task is to categorize a given text among C given classes
 - agriculture, computer chips, energy, environment, **sports, politics, gossip** *etc.*
 - friends, family, **spam, advertisements, newsletters** *etc.*

- Some documents appear more frequently than others.

$$p_{\text{cat}}(\text{gossip}) > p_{\text{cat}}(\text{philosophy})$$

Text classification & probabilistic framework

- Our goal will be to understand better the relationship between

TEXT $\overset{?}{\leftrightarrow}$ CATEGORY

- Here, we assume also that there is a **joint** probability on texts and their category.

$$P(\text{text}, \text{category})$$

which quantifies how likely is

a text `text` and a category `category` to match

- For instance,

$$P(\text{'I am feeling hungry these days'}, \text{'poetry'}) \approx 0$$

$$P(\text{'Manchester United's stock rose after their victory'}, \text{'business'})$$

∨

$$P(\text{'Manchester United's stock rose after their victory'}, \text{'sports'})$$

Text classification & probabilistic framework

- Hence, given a sequence of words (including punctuation)

$$\mathbf{w} = (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, \dots, w_n)$$

- Assuming we know P , the **joint** probability between texts and categories,
- An easy way to guess the category of \mathbf{w} is by looking at

$$\text{category-prediction}(\mathbf{w}) = \underset{C}{\operatorname{argmax}} P(C | w_1, w_2, \dots, w_n)$$

Bayes Rule

- Using Bayes theorem $p(A, B) = p(A|B)p(B)$,

$$P(\mathbf{C} | \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n) = \frac{P(\mathbf{C}, \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)}{P(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)}$$

- When looking for the category C that best fits \mathbf{w} , we only focus on the numerator.
- Bayes theorem also gives that

$$\begin{aligned} P(\mathbf{C}, \mathbf{w}_1, \dots, \mathbf{w}_n) &= P(\mathbf{C})P(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n | \mathbf{C}) \\ &= P(\mathbf{C})P(\mathbf{w}_1 | \mathbf{C})P(\mathbf{w}_2, \mathbf{w}_3, \dots, \mathbf{w}_n | \mathbf{C}, \mathbf{w}_1) \\ &= P(\mathbf{C})P(\mathbf{w}_1 | \mathbf{C})P(\mathbf{w}_2 | \mathbf{C}, \mathbf{w}_1)P(\mathbf{w}_3, \mathbf{w}_4, \dots, \mathbf{w}_n | \mathbf{C}, \mathbf{w}_1, \mathbf{w}_2) \\ &= \prod_{i=1}^n P(\mathbf{w}_i | \mathbf{C}, \mathbf{w}_1, \dots, \mathbf{w}_{i-1}) \end{aligned}$$

Examples

- Assume we have the beginning of this news title

$w_1, \dots, w_{12} =$ 'The weather was so bad that the organizers decided to close the'

- If $C =$ business, then

$$P(W_{13} = \text{'market'} | \text{business}, w_1, \dots, w_{12})$$

should be quite high, as well as *summit, meeting etc..*

- On the other hand, if we know that $C =$ sports, the probability for w_{13} changes significantly...

$$P(W_{13} = \text{'game'} | \text{sports}, w_1, \dots, w_{12})$$

The Naive Bayes Assumption

- From a factorization

$$P(\mathbf{C}, w_1, \dots, w_n) = \prod_{i=1}^n P(w_i | \mathbf{C}, w_1, \dots, w_{i-1})$$

which handles all the **conditional** structures of text,

- we assume that each word appears **independently conditionally to C** ,

$$\begin{aligned} P(w_i | \mathbf{C}, w_1, \dots, w_{i-1}) &= P(w_i | \mathbf{C}, \cancel{w_1}, \dots, \cancel{w_{i-1}}) \\ &= P(w_i | \mathbf{C}) \end{aligned}$$

- and thus

$$P(\mathbf{C}, w_1, \dots, w_n) = \prod_{i=1}^n P(w_i | \mathbf{C})$$

The Naive Bayes Assumption Leads to Word Counts

- The factorization

$$P(w_i | C, w_1, \dots, w_{i-1}) = P(w_i | C)$$

- means that we take for granted that

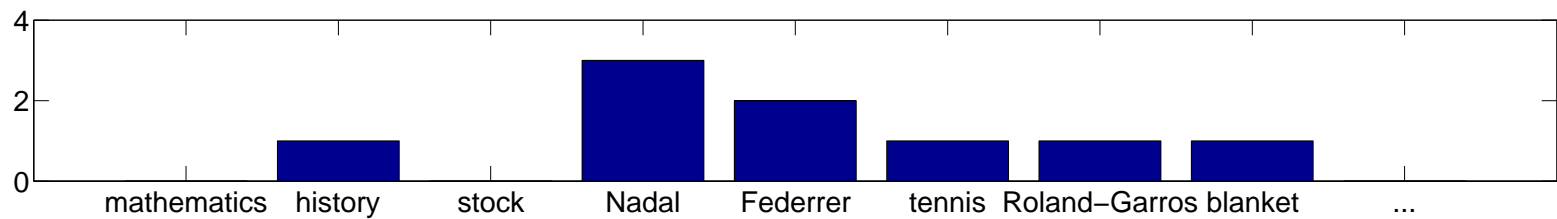
$$\begin{aligned} P(C, \text{'The weather was bad so the meeting was closed'}) \\ = \\ P(C, \text{'was The bad the closed meeting weather was so'}) \end{aligned}$$

The Naive Bayes Assumption Leads to Word Counts

- Assume we know $P(\mathcal{C}, w)$ for all words w in the dictionary and all categories.

$$P(\text{'business'}, \text{'stock'}) > P(\text{'sports'}, \text{'stock'})$$

- Given a text \mathbf{T} = But Federer has been quite a French Open security blanket for Nadal. Their rivalry is one of the greatest in tennis history, yet it has been decidedly short on suspense here. Nadal is now 5-0 against Federer at Roland-Garros. Nadal is the greatest ...
- The only thing the Bayes classifier will consider is the **word histogram**



The Naive Bayes Assumption Leads to Word Counts

- To each text,
 - count the frequency of each word w in the dictionary \mathcal{D} , h_w . Then

$$P(\mathbf{T}|\mathbf{C}) = \prod_{w \in \mathcal{D}} P(w|\mathbf{C})^{h_w}$$

- In the example below, it seems obvious that the terms

$$P(W = \text{'Nadal'}|\text{tennis}), P(W = \text{'Federer'}|\text{tennis}), \dots$$

will be quite big.

- The Naive Bayes should easily classify this text as tennis...
 - **if the probabilities $P(w|\mathbf{C})$ were known!!!**

Term Frequencies

In theory, we need to build an estimate of $P(w|\mathbf{C})$ for **all** words of \mathcal{D} , all categories

A typical approach

- Consider a corpus of documents with different categories of text $\{(\mathbf{T}_1, c_1), \dots, (\mathbf{T}_N, c_N)\}$.
- Build a reduced dictionary $\hat{\mathcal{D}}$
 - using **all** words appearing in all \mathbf{T}_i ,
 - usually removing non-informative words such as articles, prepositions *etc.*
- Compute histograms h_w^i for **each** \mathbf{T}_i which only track words in $\hat{\mathcal{D}}$.
- Compute an estimate $\hat{p}(w|c)$ for each word $w \in \hat{\mathcal{D}}$ and estimates $\hat{p}(c)$.

Term Frequencies

- Use these elements, \hat{p} , $\hat{\mathcal{D}}$ to classify a new text \mathbf{T} using his representation $h_w^{\mathbf{T}}$

$$\text{category-prediction}(\mathbf{T}) = \underset{c}{\operatorname{argmax}} \left(\hat{p}(c) \prod_{w \in \hat{\mathcal{D}}} \hat{p}(w|c)^{h_w^{\mathbf{T}}} \right)$$

- of course, if we use the logarithm of the r.h.s., we get the rule

$$\text{category-prediction}(\mathbf{T}) = \underset{c}{\operatorname{argmax}} \log \hat{p}(c) + \sum_{w \in \hat{\mathcal{D}}} h_w^{\mathbf{T}} \log \hat{p}(w|c)$$

Naive Bayes for text \Leftrightarrow Linear Classifier Using Term Frequencies as Features

- Once this is established... we could imagine **any** linear classifier using TF.

Term Frequency Data Seen from a Classification Perspective

- The **Data** we have:
 - texts \mathbf{T}_i translated as histograms of words $h^1, h^2, h^3, \dots, h^N$.
 - Each histogram is a vector of the simplex Σ_d where $d = \#\mathcal{D} - 1$ and

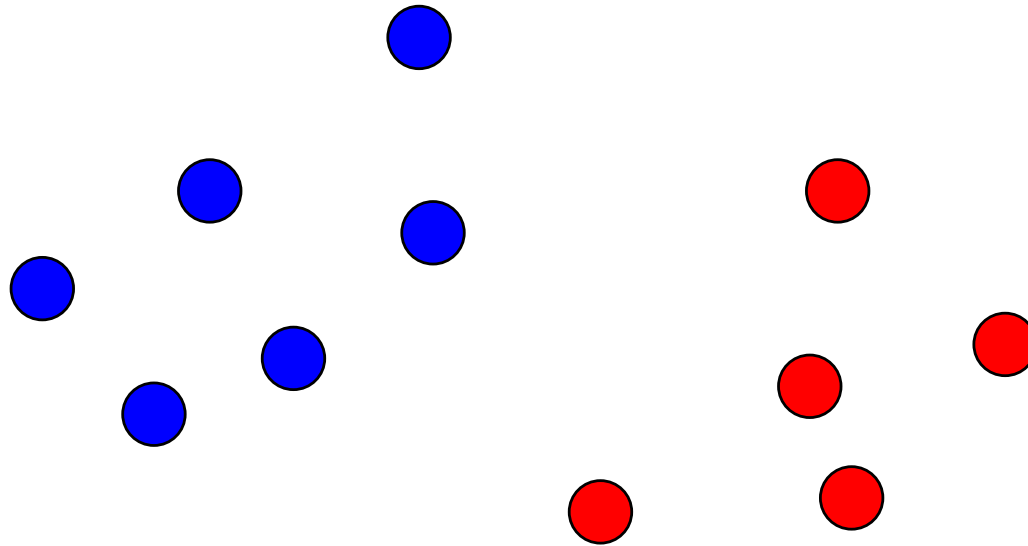
$$\Sigma_d = \left\{ x \in \mathbb{R}^{d+1} \mid x_i \geq 0, \sum_{i=1}^{d+1} x_i = 1 \right\}.$$

- We consider 2 categories only here, for instance “spam” vs “non-spam”.
- The corpus consists in a large number of **histogram/bit** pairs

$$\text{“training set”} = \left\{ \left(h_i = \begin{bmatrix} h_{w_1}^i \\ h_{w_2}^i \\ \vdots \\ h_{w_{d+1}}^i \end{bmatrix} \in \Sigma^d, \mathbf{y}_i \in \{0, 1\} \right)_{i=1..N} \right\}$$

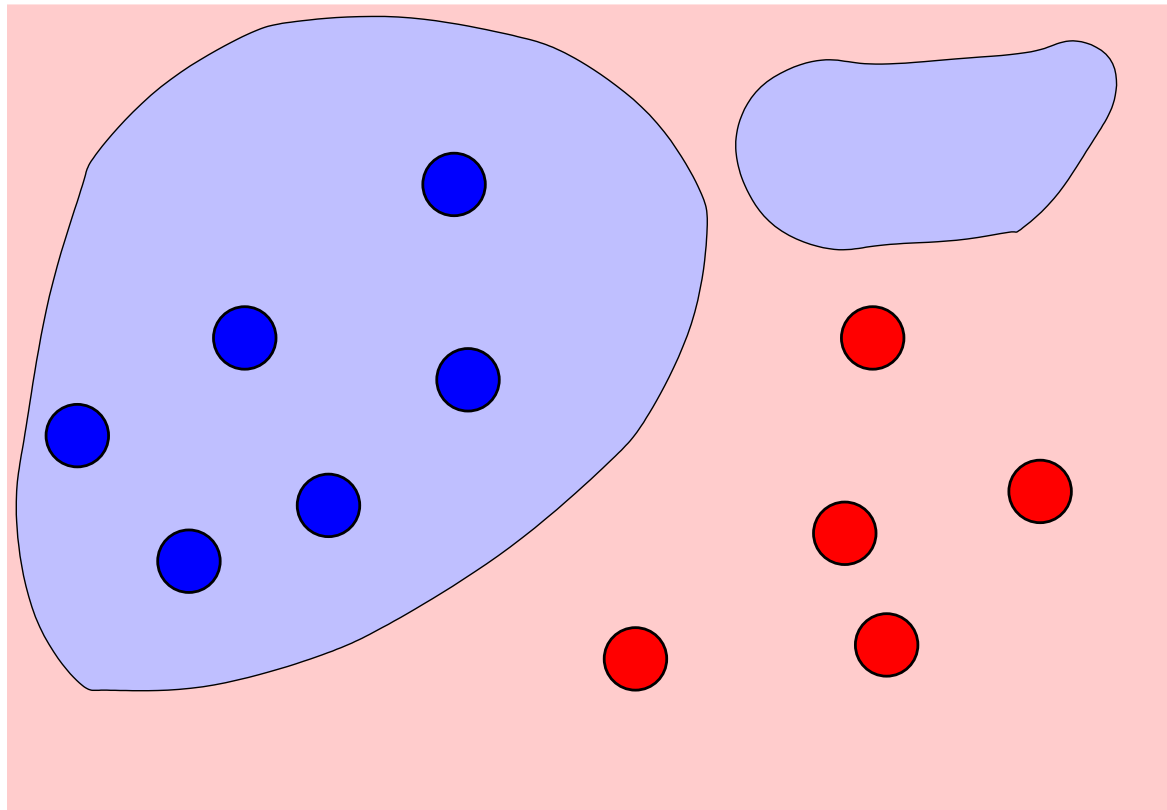
- For illustration purposes **only** we will consider the 2 dimensional simplex, that is $\#\mathcal{D} = 3$.

Binary Classification Separation Surfaces for Vectors



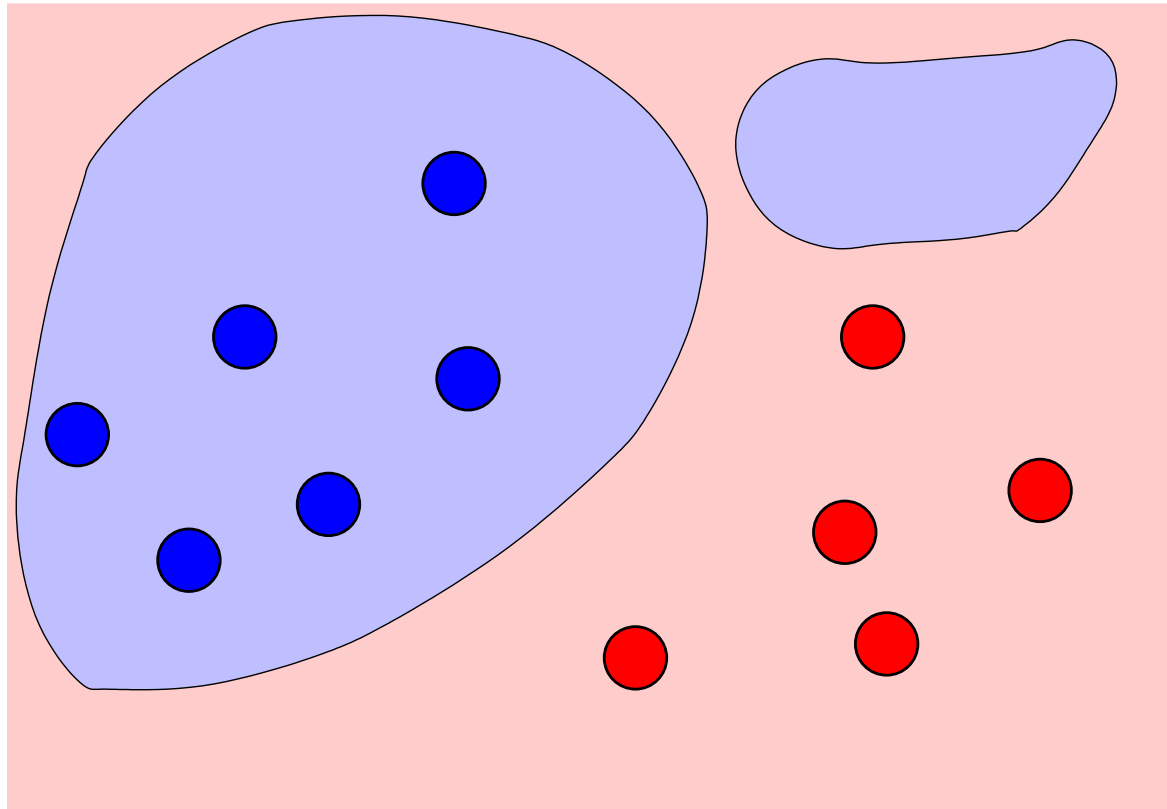
What is a classification rule?

Binary Classification Separation Surfaces for Vectors



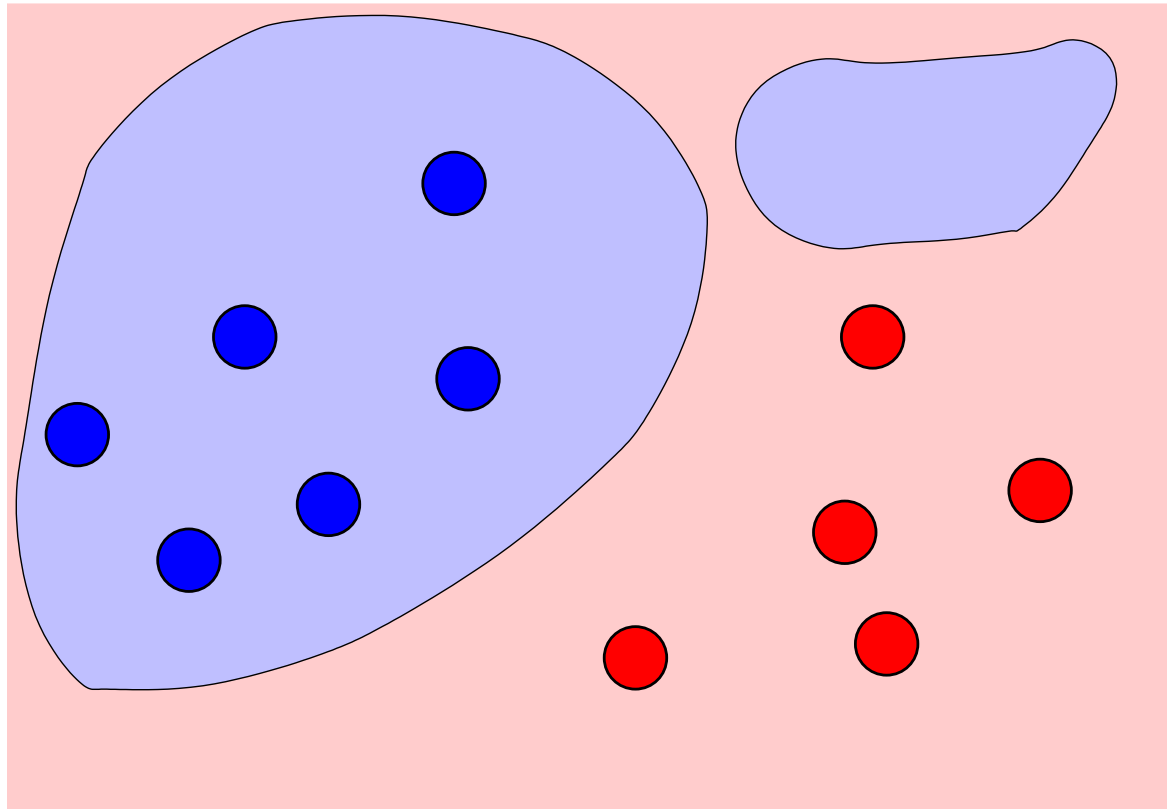
Classification rule = a partition of \mathbb{R}^d into two sets

Binary Classification Separation Surfaces for Vectors



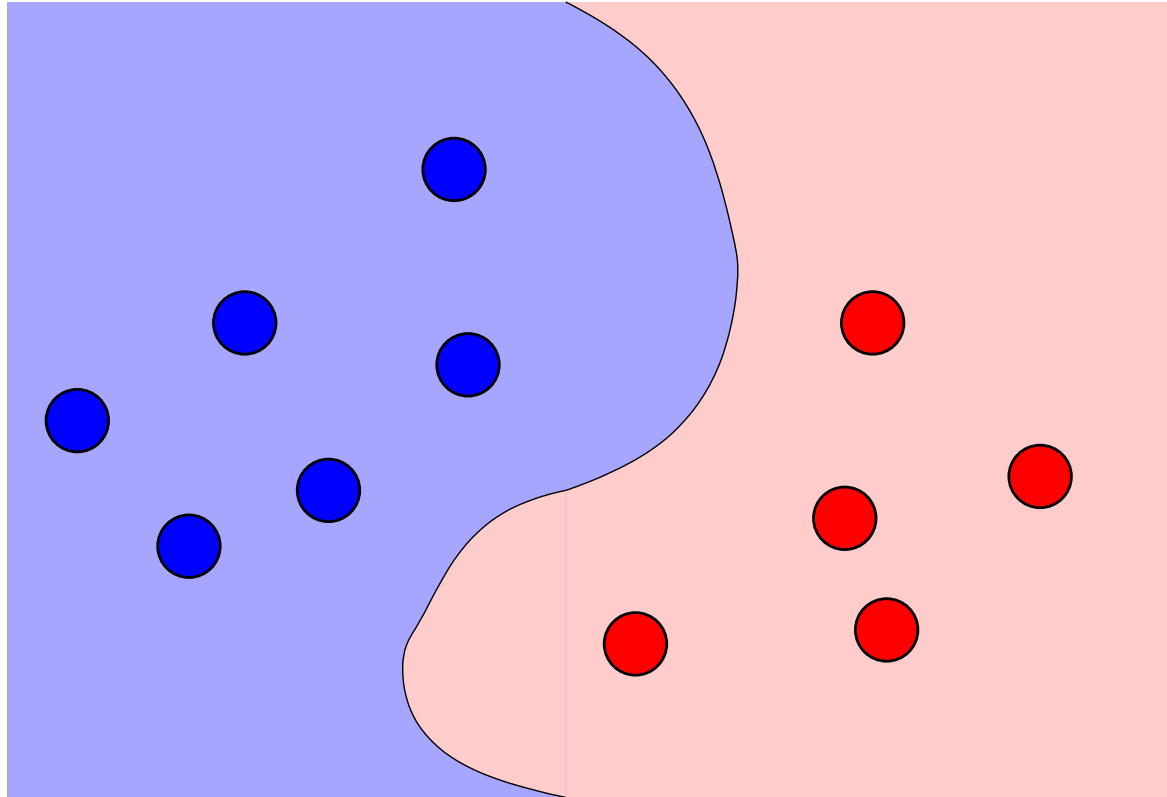
This partition is usually interpreted as the level set of a function

Binary Classification Separation Surfaces for Vectors



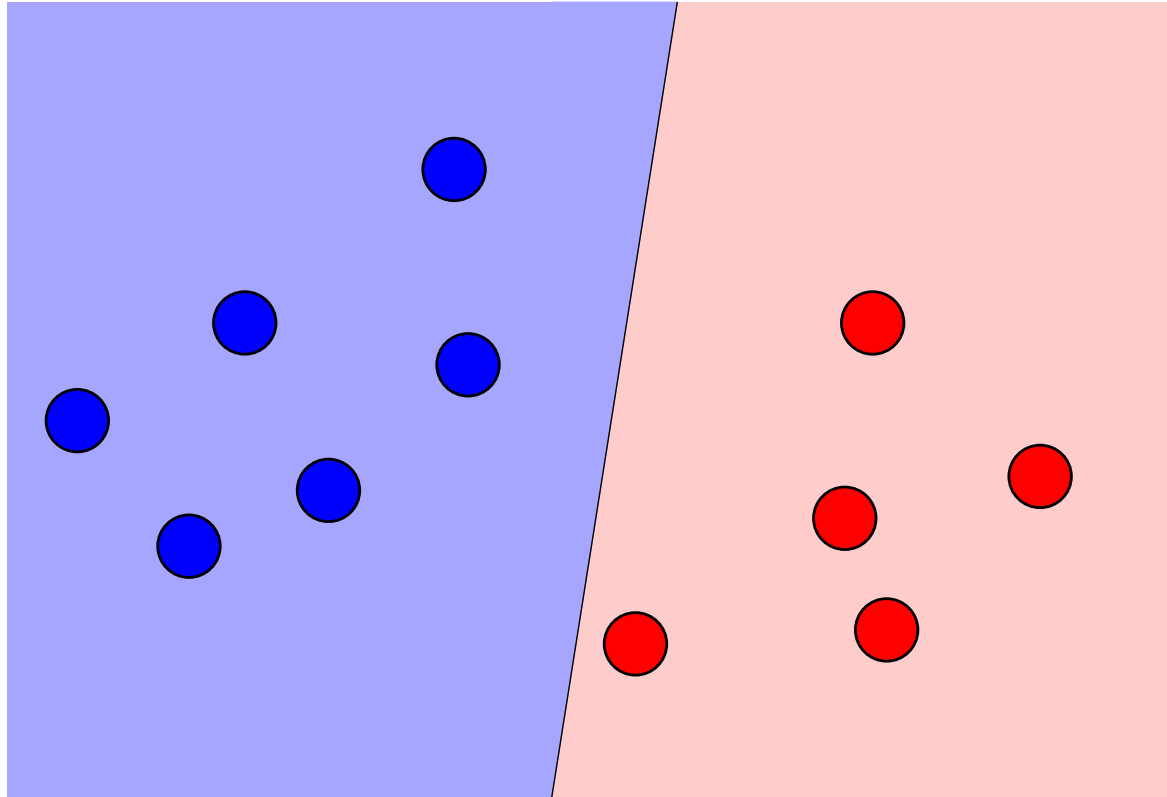
Typically, $\{h \in \Sigma_d | \mathbf{f}(h) > 0\}$ and $\{h \in \Sigma_d | \mathbf{f}(h) \leq 0\}$

Classification Separation Surfaces for Vectors



Can be defined by a single surface, *e.g.* a curved line

Classification Separation Surfaces for Vectors



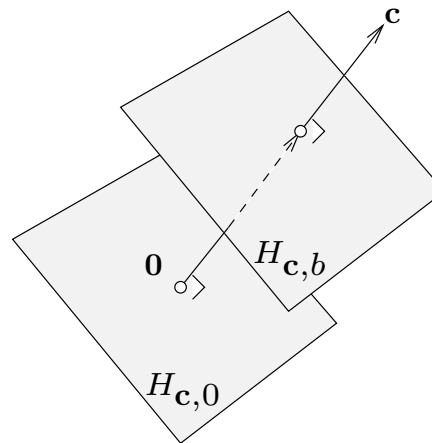
Even more **simple**: using **straight lines** and halfspaces.

Linear Classifiers

- **Straight lines** (hyperplanes when $d > 2$) are **the simplest type** of classifiers.
- A hyperplane $H_{\mathbf{c},b}$ is a set in \mathbb{R}^p defined by
 - a normal vector $\mathbf{c} \in \mathbb{R}^p$
 - a constant $b \in \mathbb{R}$. as

$$H_{\mathbf{c},b} = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{c}^T \mathbf{x} = b\}$$

- Letting b vary we can “slide” the hyperplane across \mathbb{R}^p

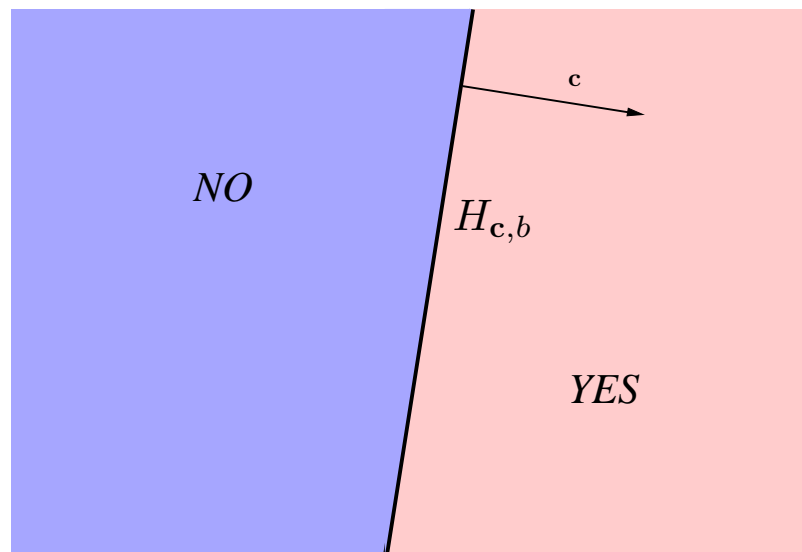


Linear Classifiers

- In Σ_d , things hypersurfaces **divide** \mathbb{R}^d into **two** halfspaces,

$$\{h \in \mathbb{R}^d \mid \mathbf{c}^T h < b\} \cup \{h \in \mathbb{R}^d \mid \mathbf{c}^T h \geq b\} = \mathbb{R}^d$$

- Linear classifiers attribute the “yes” and “no” answers given arbitrary \mathbf{c} and b .



- Assuming we only look at halfspaces for the decision surface...
...how to **choose the “best”** (\mathbf{c}^*, b^*) given a training sample?

Linear Classifiers

- This specific question,

“training set” $\{(h^i \in \Sigma^d, \mathbf{y}_i \in \{0, 1\})_{i=1..N}\} \xrightarrow{????}$ “best” \mathbf{c}^*, b^*

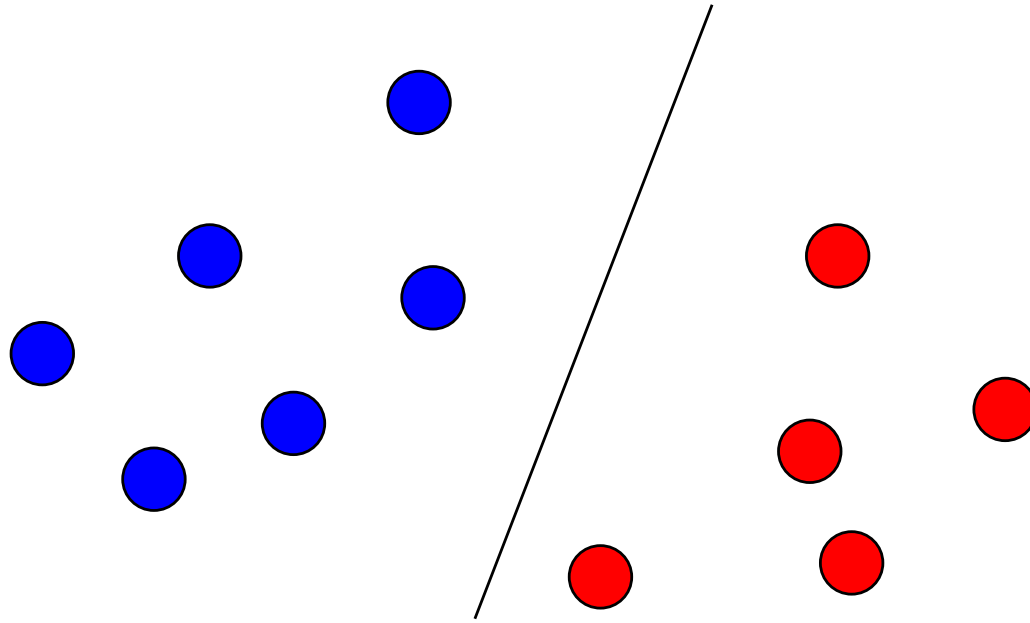
has different answers. Depends on the meaning of “best” ?:

- **Linear Discriminant Analysis** (or Fisher’s Linear Discriminant);
- **Logistic regression** maximum likelihood estimation;
- **Perceptron**, a one-layer neural network;
- **Support Vector Machine**, the result of a convex program
- *etc.*

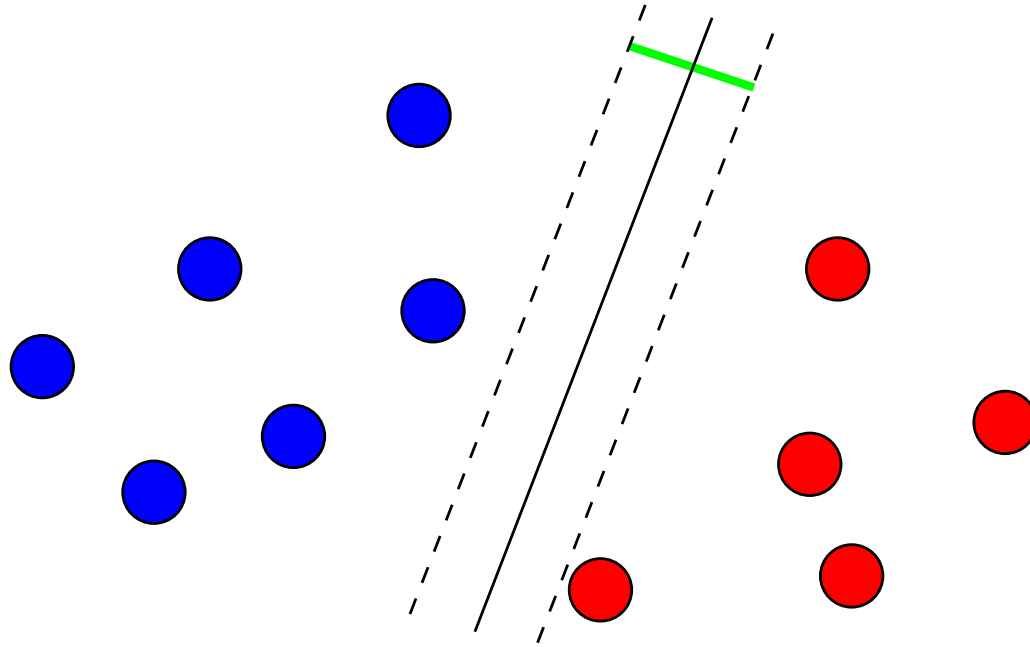
Support Vector Machine

Check the very nice book on the subject by T.Joachims. It's a bit old now but contains a lot of fundamental ideas.

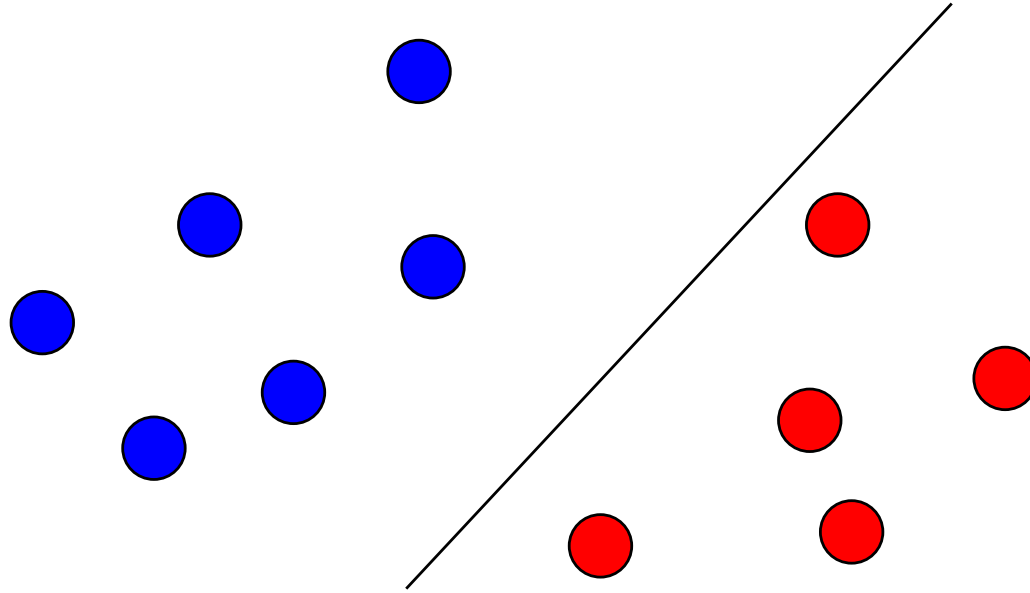
A criterion to select a linear classifier: the margin ?



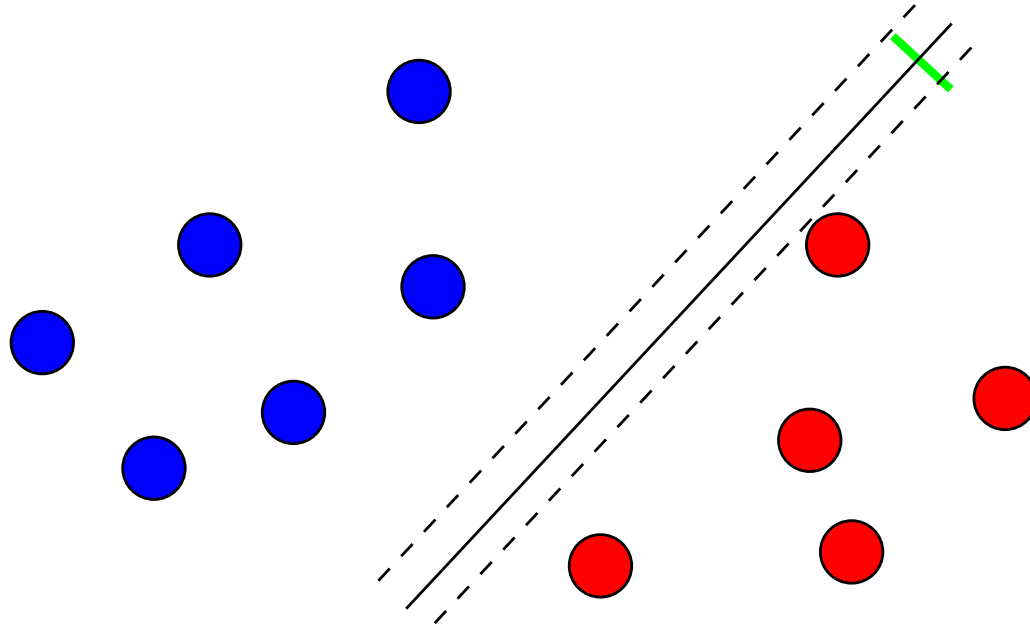
A criterion to select a linear classifier: the margin ?



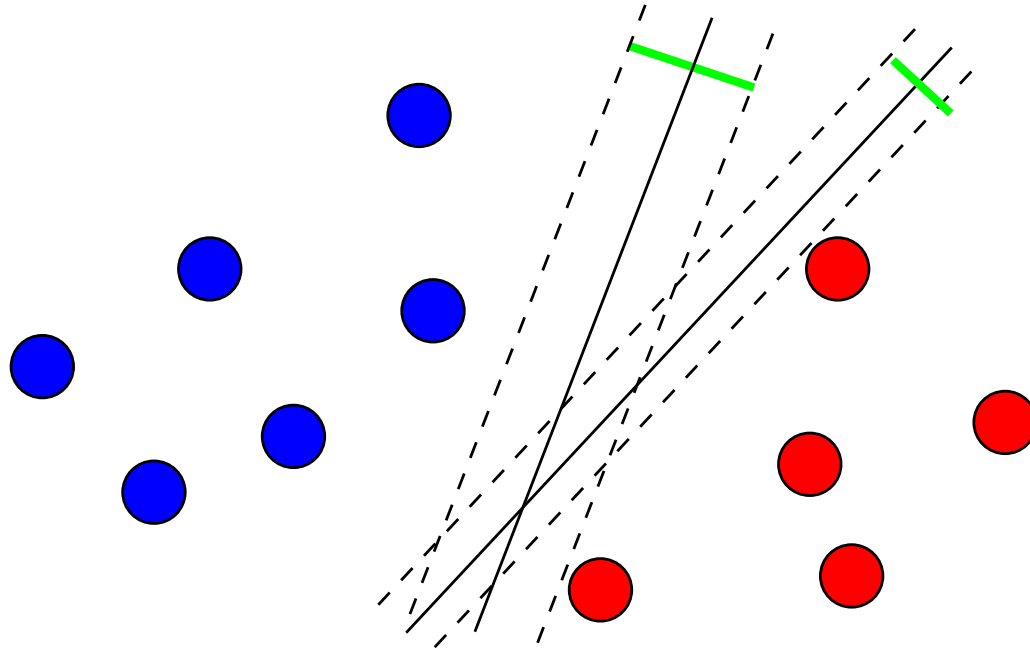
A criterion to select a linear classifier: the margin ?



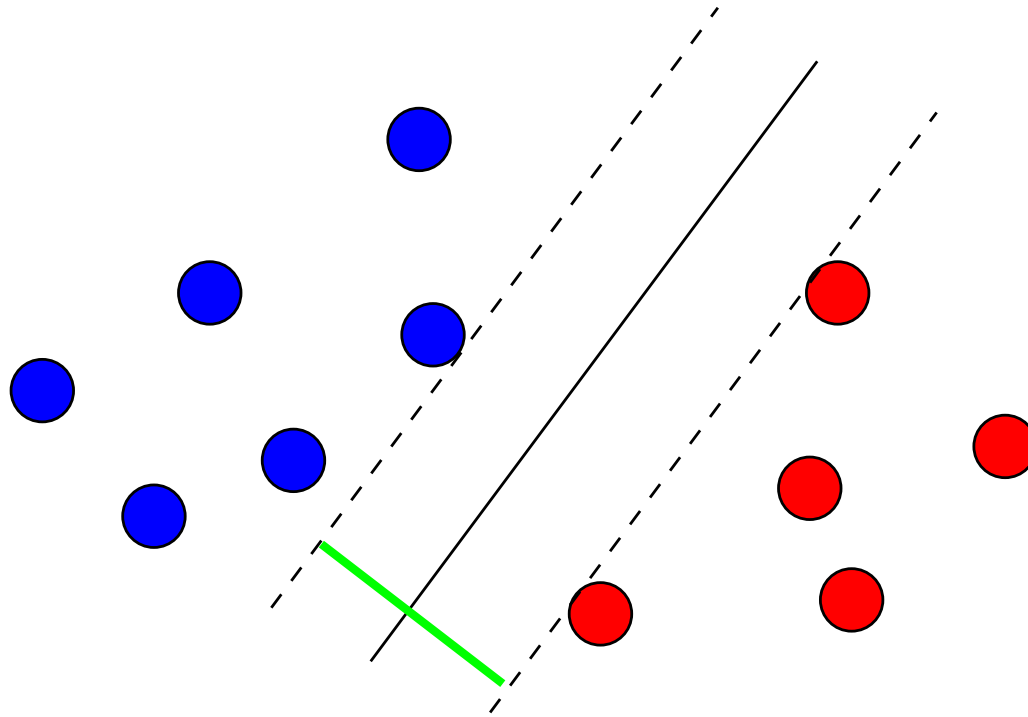
A criterion to select a linear classifier: the margin ?



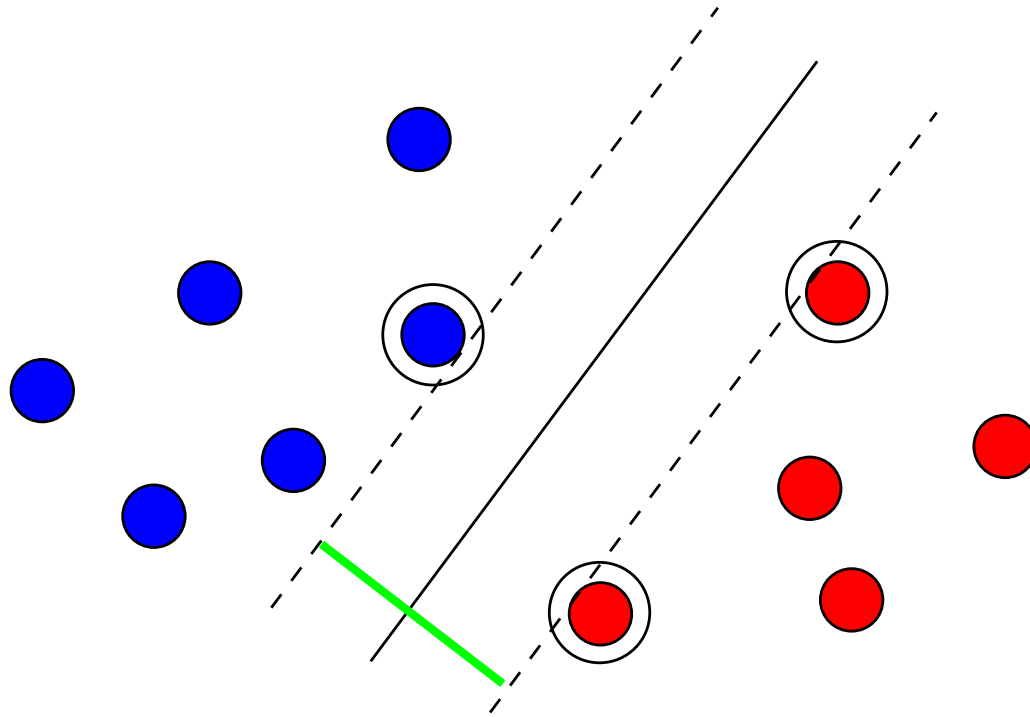
A criterion to select a linear classifier: the margin ?



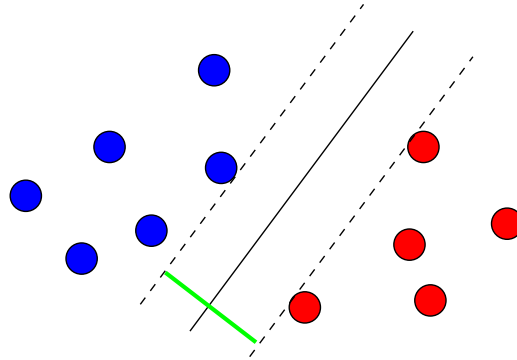
Largest Margin Linear Classifier ?



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:

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

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:

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

- 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 $\alpha_i \geq 0$)

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

The Lagrange dual function

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

has saddle points when

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

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

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. KKT gives us $\alpha_i (\mathbf{y}_i (\mathbf{w}^T \mathbf{x}_i + b) - 1) = 0$,
...hence, either **$\alpha_i = 0$** or **$\mathbf{y}_i (\mathbf{w}^T \mathbf{x}_i + b) = 1$** .
- $\alpha_i \neq 0$ **only** for points on the support hyperplanes $\{(\mathbf{x}, \mathbf{y}) \mid \mathbf{y}_i (\mathbf{w}^T \mathbf{x}_i + b) = 1\}$.

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 $\alpha_i > 0$, $\mathbf{y}_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$,

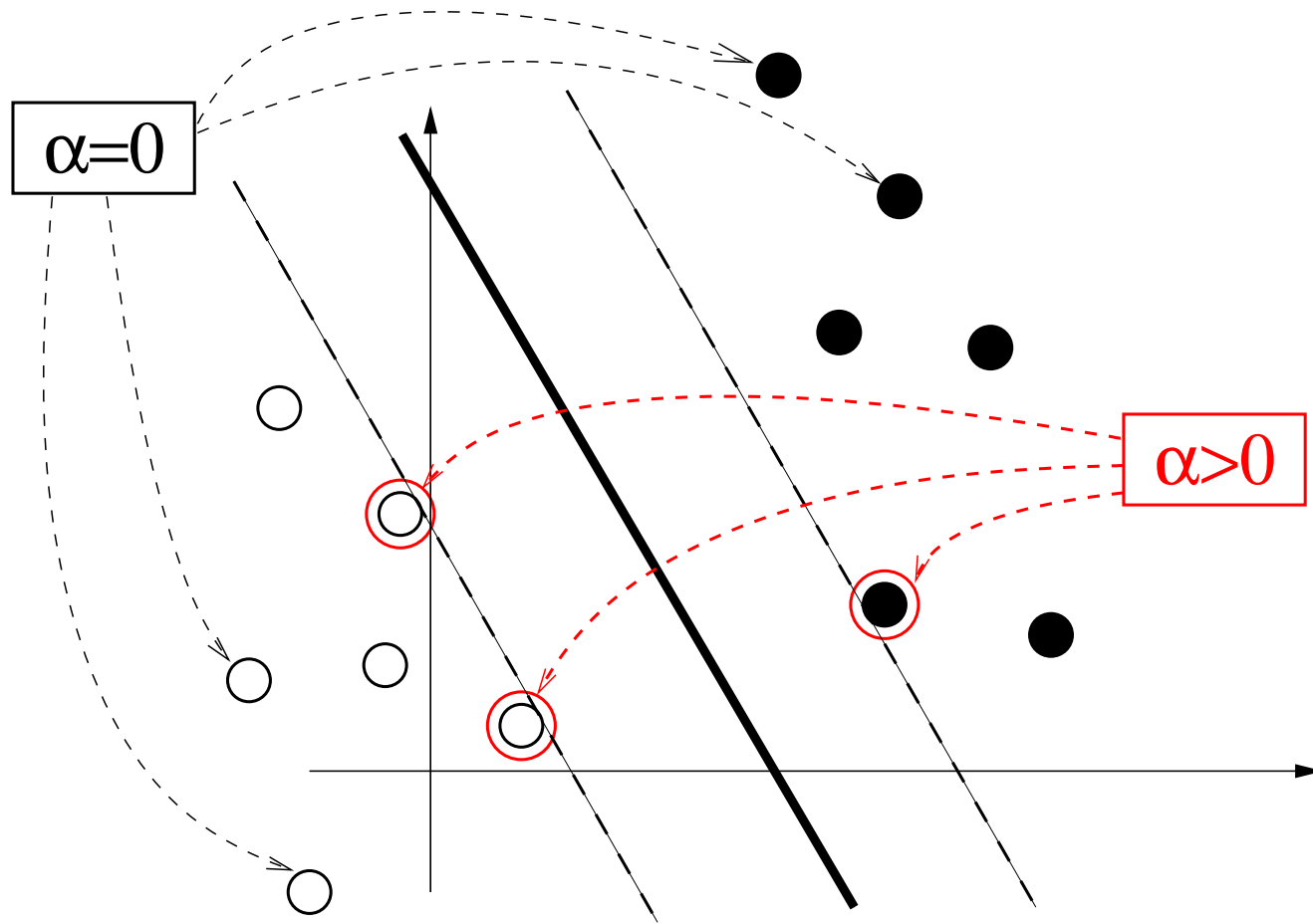
$$b^* = -\frac{1}{2} \left(\min_{\mathbf{y}_i=1, \alpha_i>0} (\mathbf{w}^T \mathbf{x}_i) + \max_{\mathbf{y}_i=-1, \alpha_i>0} (\mathbf{w}^T \mathbf{x}_i) \right)$$

- the **decision function** is therefore:

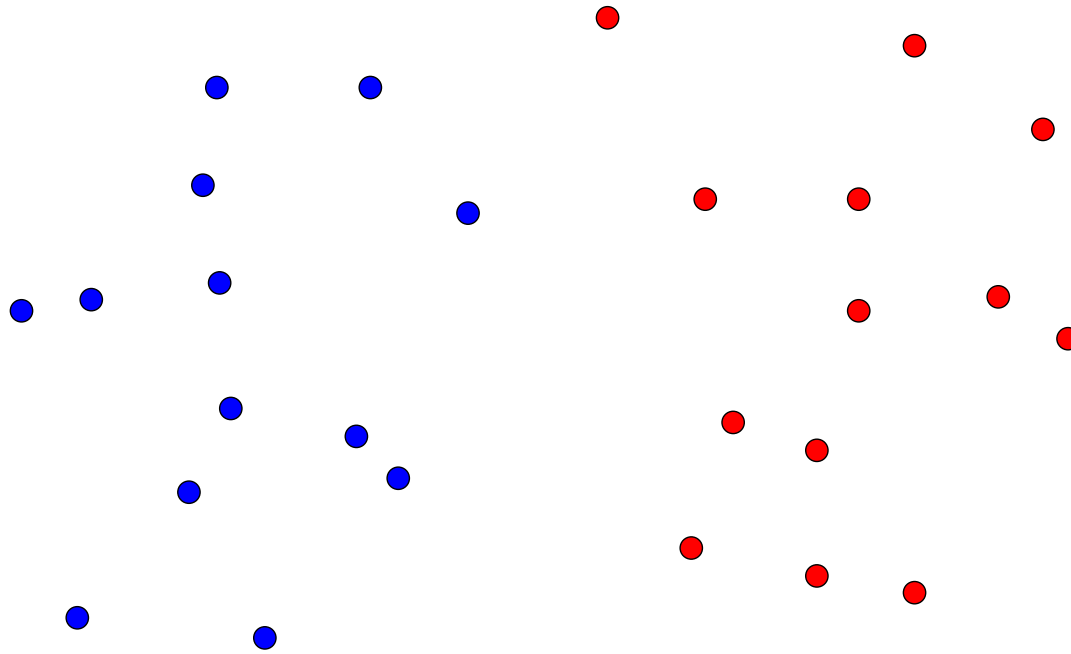
$$\begin{aligned} f^*(\mathbf{x}) &= \mathbf{w}^T \mathbf{x} + b^* \\ &= \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b^*. \end{aligned}$$

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

Interpretation: support vectors

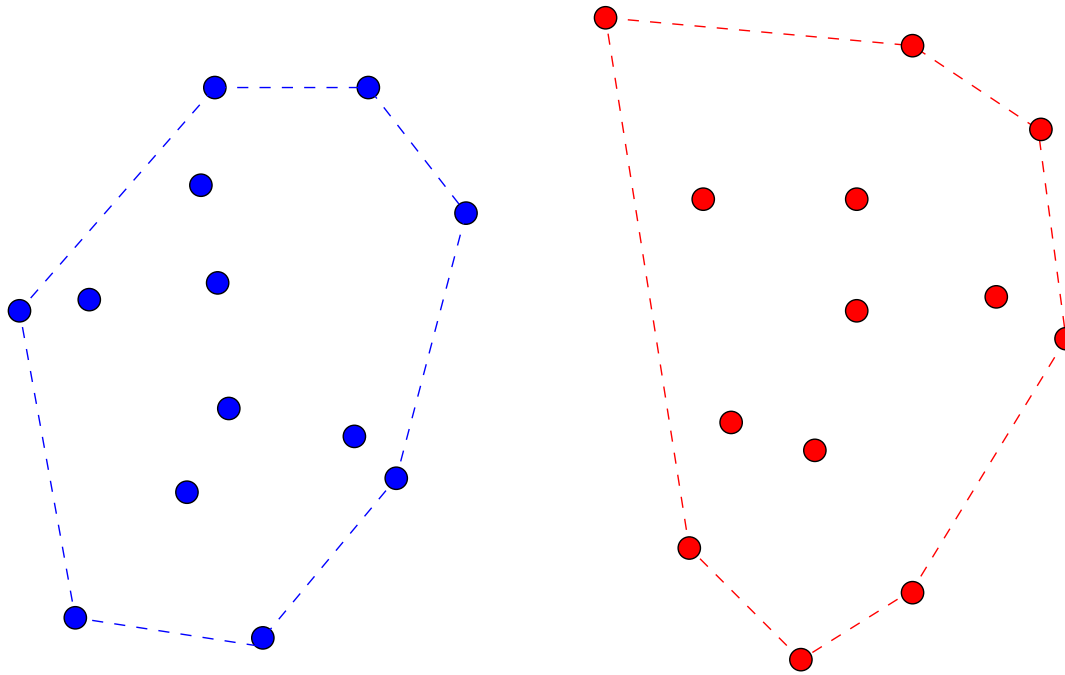


Another interpretation: Convex Hulls



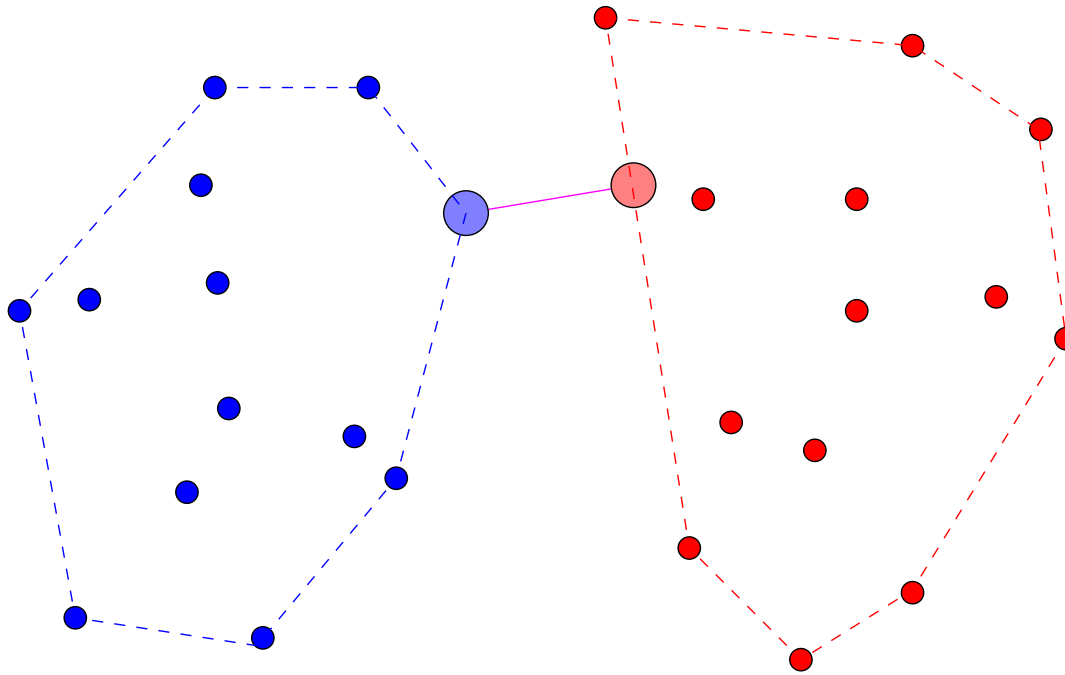
go back to 2 sets of points that are linearly separable

Another interpretation: Convex Hulls



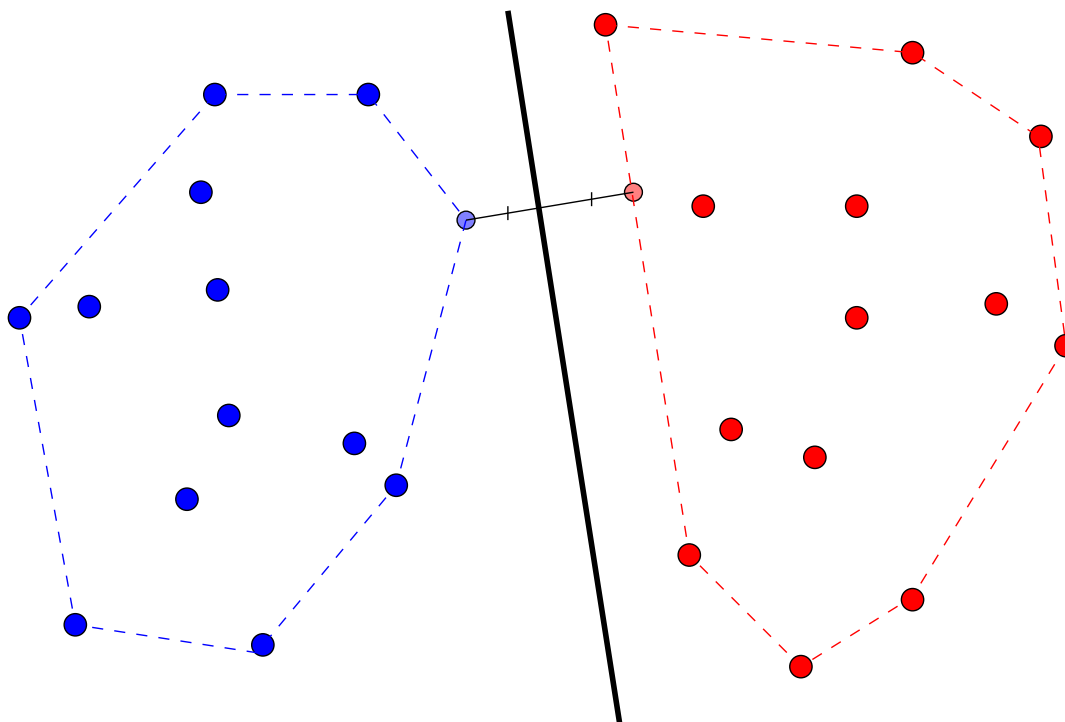
Linearly separable = convex hulls do not intersect

Another interpretation: Convex Hulls



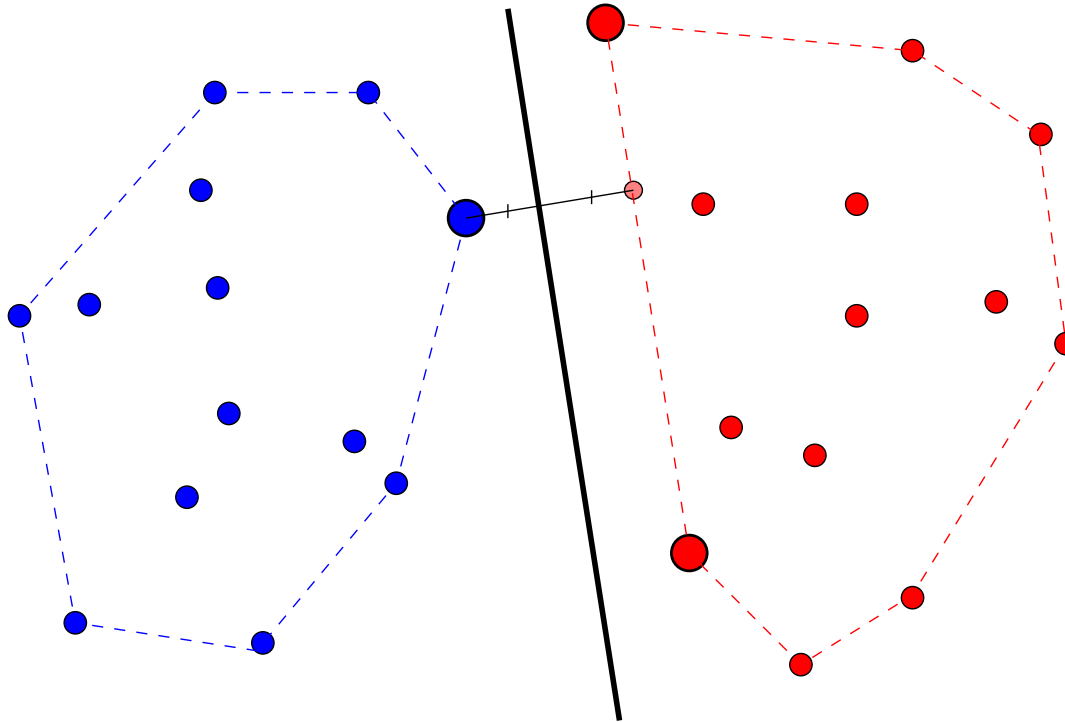
Find two closest points, one in each convex hull

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

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

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

under the constraints:

$$\begin{cases} 0 \leq \alpha_i \leq C, & \text{for } i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0. \end{cases}$$

- The **decision function** becomes:

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

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 \mathbf{K} and \mathbf{y} :

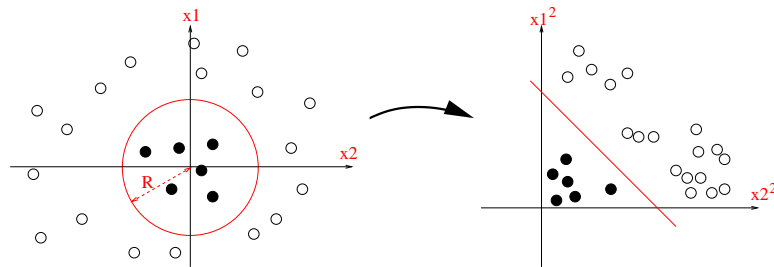
$$\begin{aligned} &\text{maximize} && g(\alpha) = \alpha^T \mathbf{1} - \frac{1}{2} \alpha^T (\mathbf{K} \odot \mathbf{y}\mathbf{y}^T) \alpha \\ &\text{such that} && 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \dots, n \\ &&& \sum_{i=1}^n \alpha_i \mathbf{y}_i = 0. \end{aligned}$$

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

Kernel example: polynomial kernel

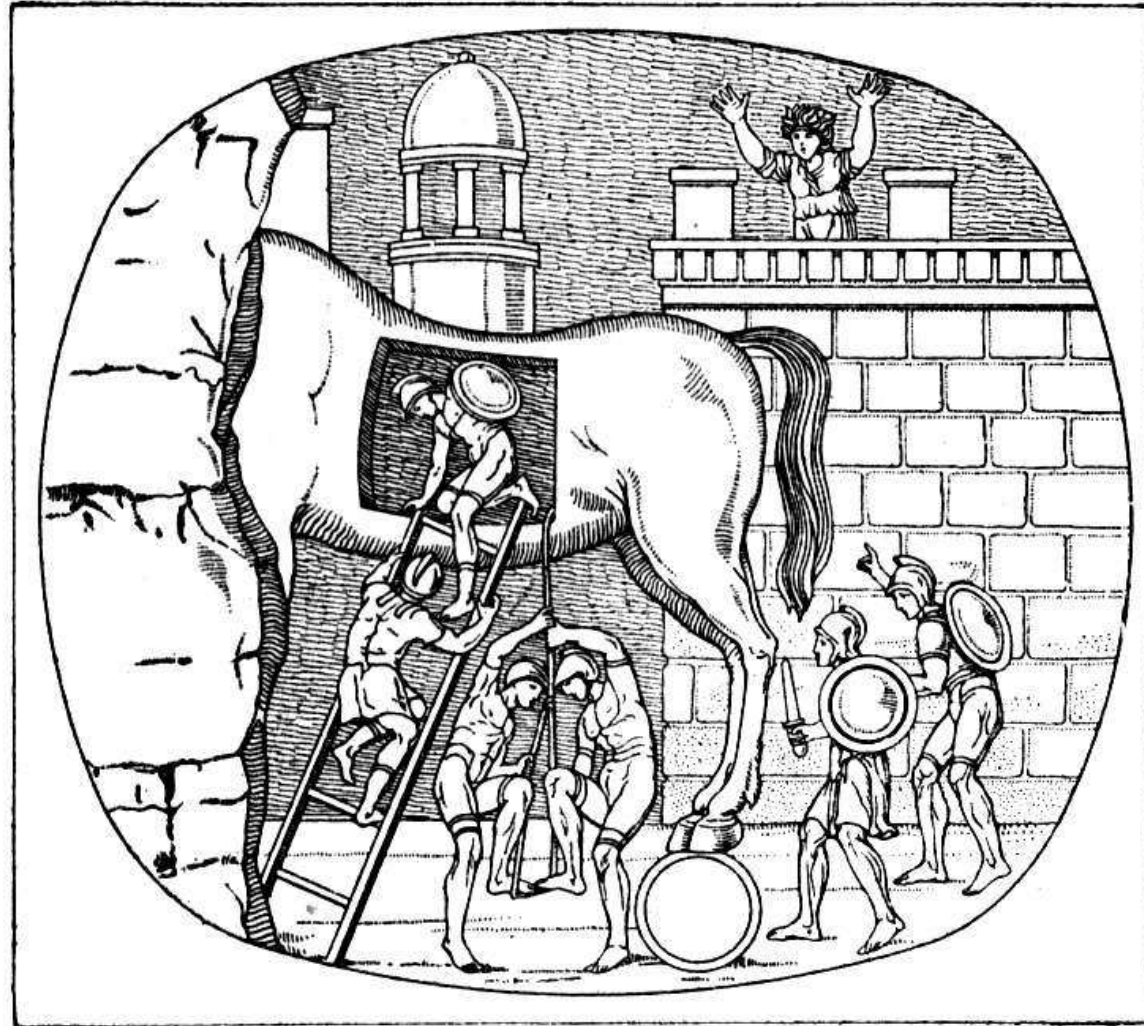
- For $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$, let $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$:

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



Kernels are Trojan Horses onto Linear Models

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



Kernels For Histograms

- 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

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