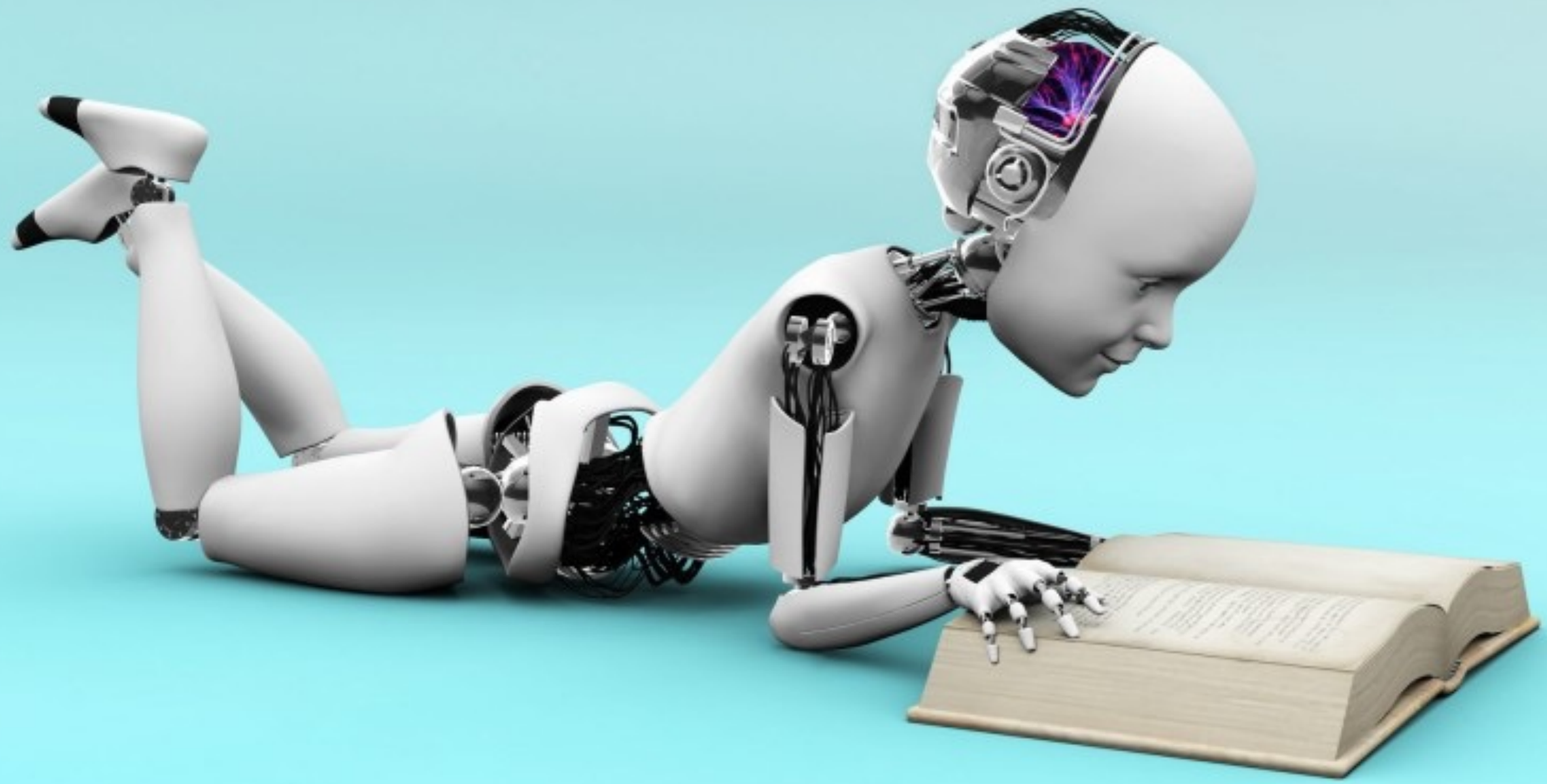
Crash Course on Machine Learning

August 22–24, 2016



simula
education

simula

Regularisation II. Dimensionality Reduction

Variable Selection, Sparsity, Clustering

Crash Course on Machine Learning

August 22-24, 2016

Valeriya Naumova and Arnaud Gotlieb

Simula Research Laboratory AS

Simula School of Research and Innovation (SSRI)

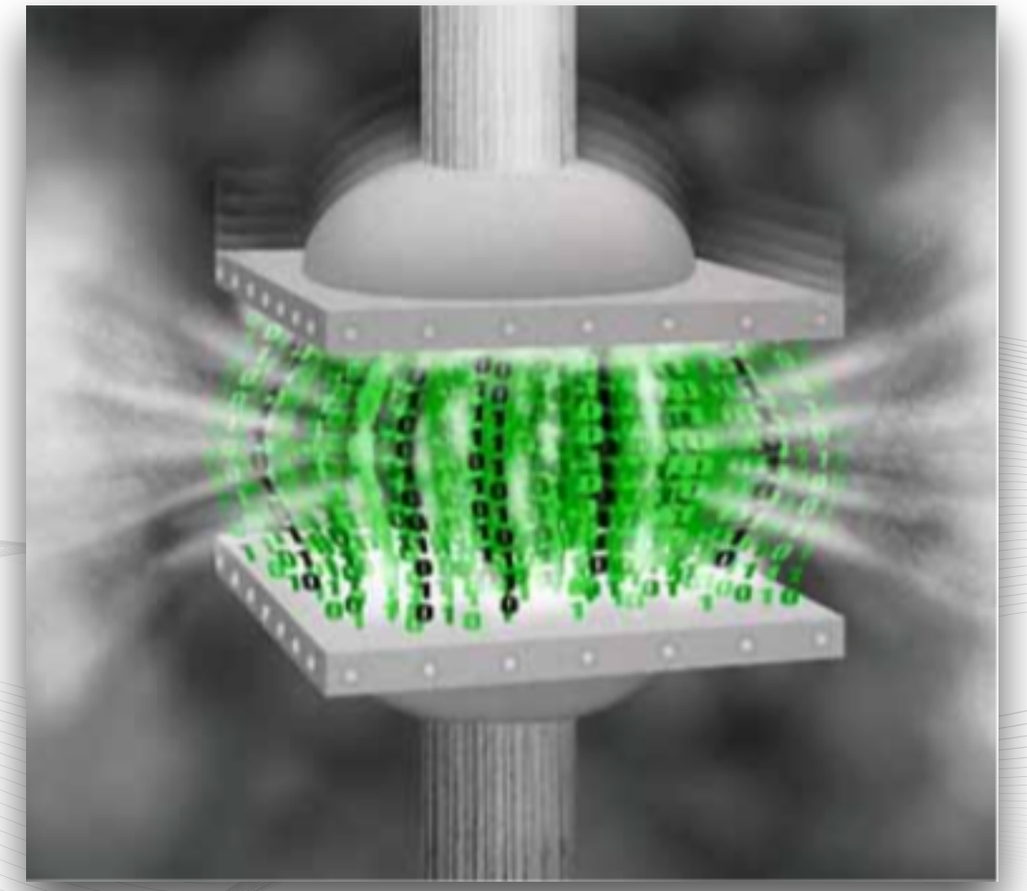
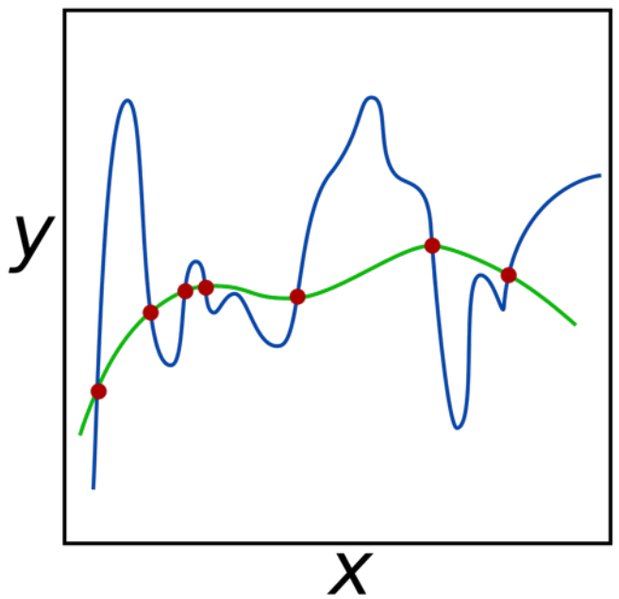
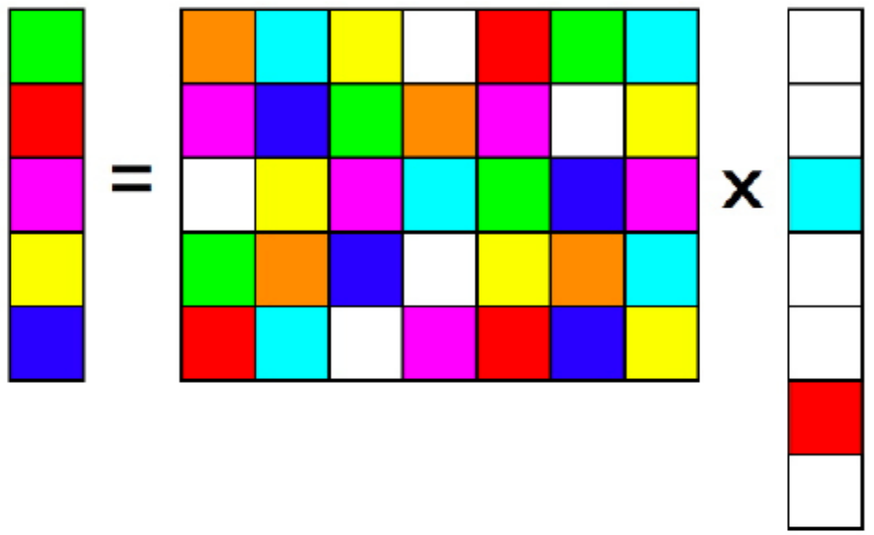


Table of Contents

Regularisation II

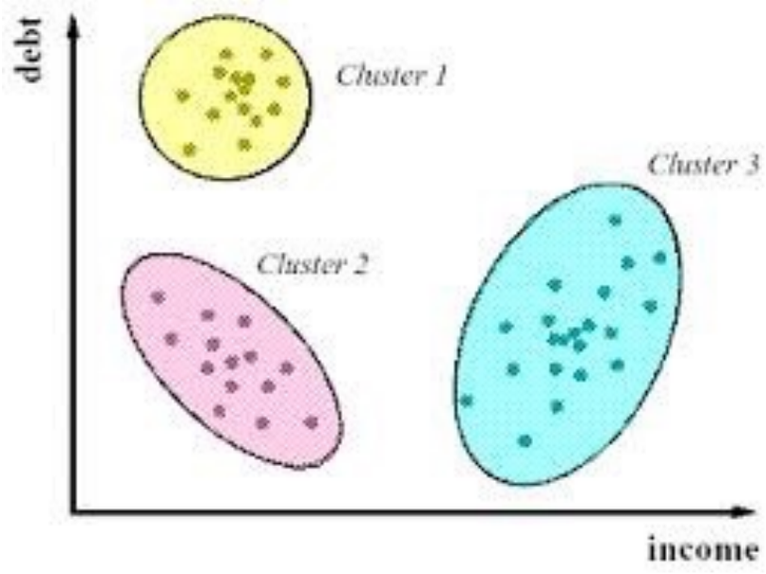


Variable Selection.
Sparsity



Dimensionality
Reduction

Clustering



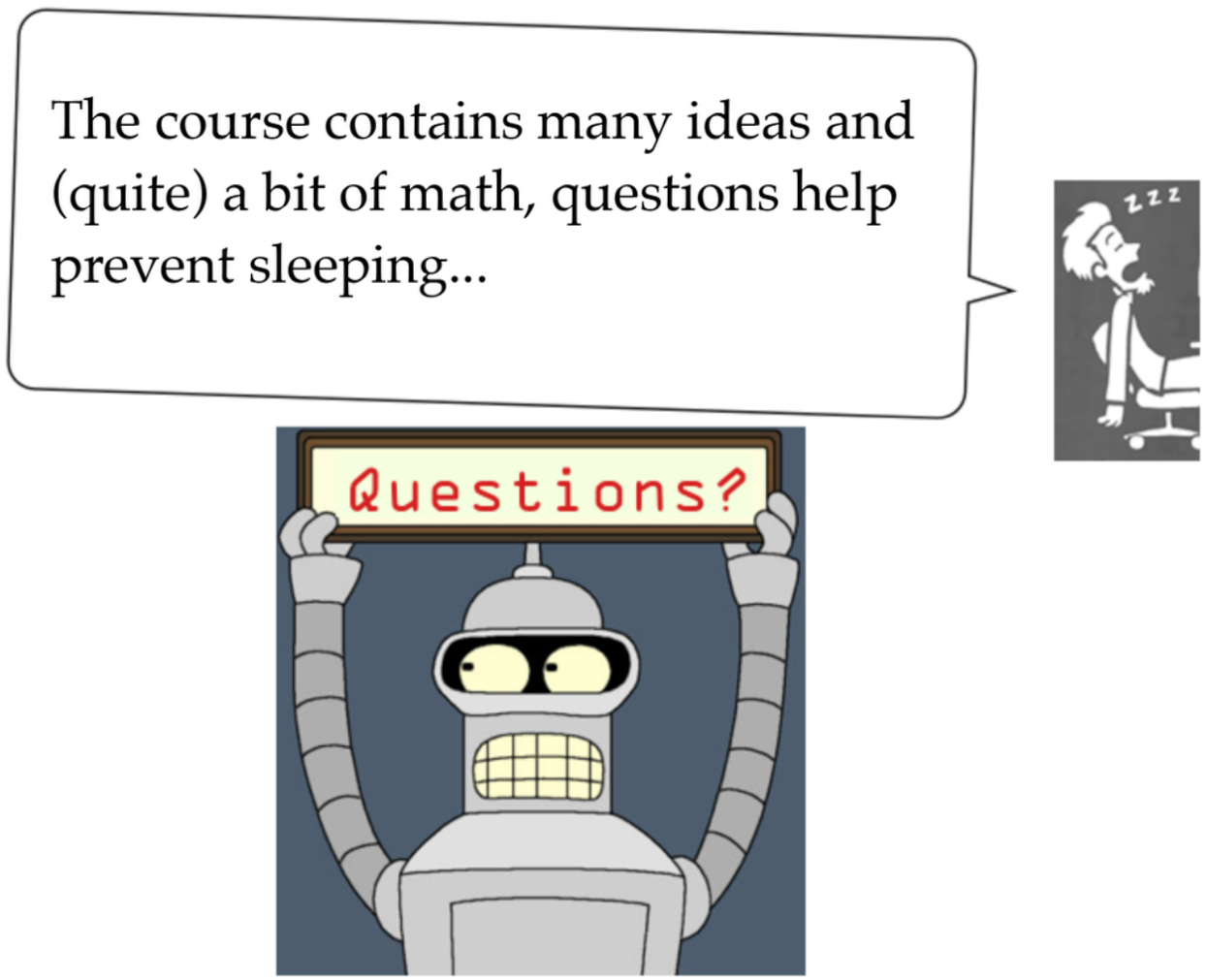
Colour Coding / Feedback

Really important stuff

Important stuff

Regular stuff

Let us know if you have comments, concerns, suggestions!



GOAL:

Consider the basic (global) regularisation methods and study their computational aspects

Going Global + Impose Smoothness

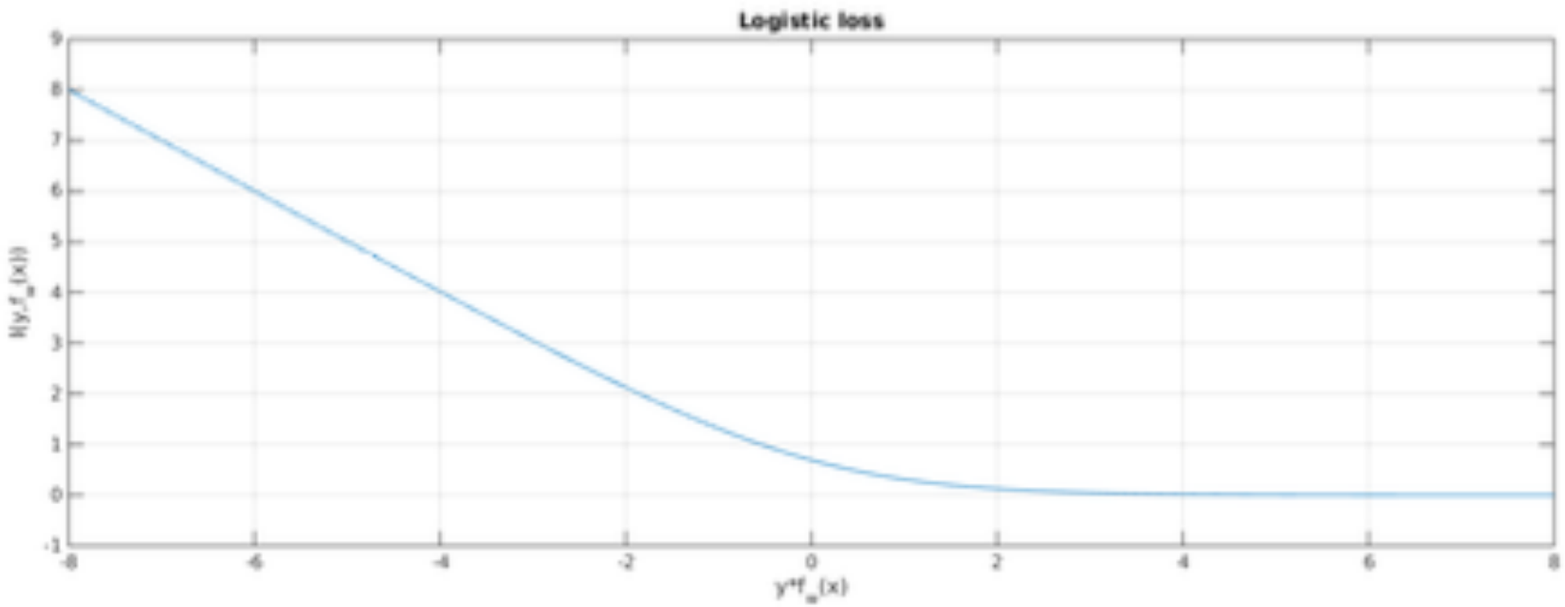
Introduction: Regularised Logistic Regression

Regularised logistic regression: Tikhonov regularisation

$$\min_{\omega \in \mathbb{R}^d} \hat{\mathcal{E}}(f) + \lambda \|\omega\|^2, \quad \hat{\mathcal{E}}(f) = \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i))$$

with the *logistic loss function:*

$$V(y, f(x)) = \log(1 + e^{-yf(x)})$$



Plot of the logistic loss function

Regularised Logistic Regression

Logistic regression minimisation

- The logistic function is differentiable.
- The candidate to compute a minimiser is the **gradient descent (GD) algorithm.**
- There regularised ERM problem associated with the logistic loss is called **regularised logistic regression**.
- Its solutions can be computed via gradient descent.

$$\nabla\hat{\mathcal{E}}(f) = \frac{1}{n}\sum_{i=1}^{n} x_i \frac{-y_i e^{-y_i x_i^T \omega_{t-1}}}{1+e^{-y_i x_i^T \omega_{t-1}}} = \frac{1}{n}\sum_{i=1}^{n} x_i \frac{-y_i}{1+e^{y_i x_i^T \omega_{t-1}}}$$

RLR: Gradient Descent Iteration

For $\omega_0 = 0$, the GD iteration applied to

$$\min_{\omega \in \mathbb{R}^d} \hat{\mathcal{E}}(f) + \lambda \|\omega\|^2$$

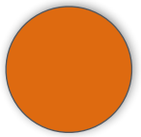
is

$$\omega_t = \omega_{t-1} - \gamma \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} x_i \frac{-y_i}{1 + e^{y_i x_i^T \omega_{t-1}}} + 2\lambda \omega_{t-1} \right)}_{a}$$

for $t = 1, \ldots, T$, where

$$a = \nabla(\hat{\mathcal{E}}(f) + \lambda \|\omega\|^2)$$

Logistic Regression and Confidence Estimation



- The solution of logistic regression has a probabilistic interpretation.
- It can be derived from the following model

$$p(1|x) = \underbrace{\frac{e^{x^T\omega}}{1+e^{x^T\omega}}}_{h}$$

where h is called logistic function.

- This can be used to compute a confidence for each prediction.

Support Vector Machines (SVM)

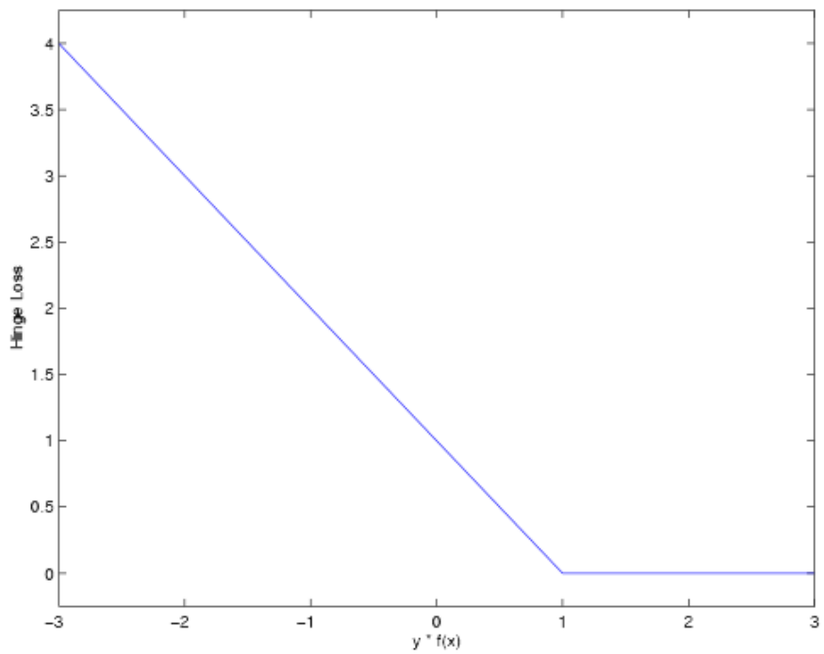
Formulation in terms of Tikhonov regularisation:

$$\min_{\omega \in \mathbb{R}^d} \hat{\mathcal{E}}(f) + \lambda \|\omega\|^2, \quad \hat{\mathcal{E}}(f) = \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i))$$

with the *Hinge loss function:*

$$V(f(x), y) = |1 - yf(x)|_+,$$

where $|s|_+ = \max(s, 0)$



Plot of the Hinge loss function

A more classical formulation (linear case)

$$\omega^* = \min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} |1 - y_i \omega^T x_i|_+ + \lambda \|\omega\|^2$$

with $\lambda = \frac{1}{C}$

Problem: non-differentiability!!!

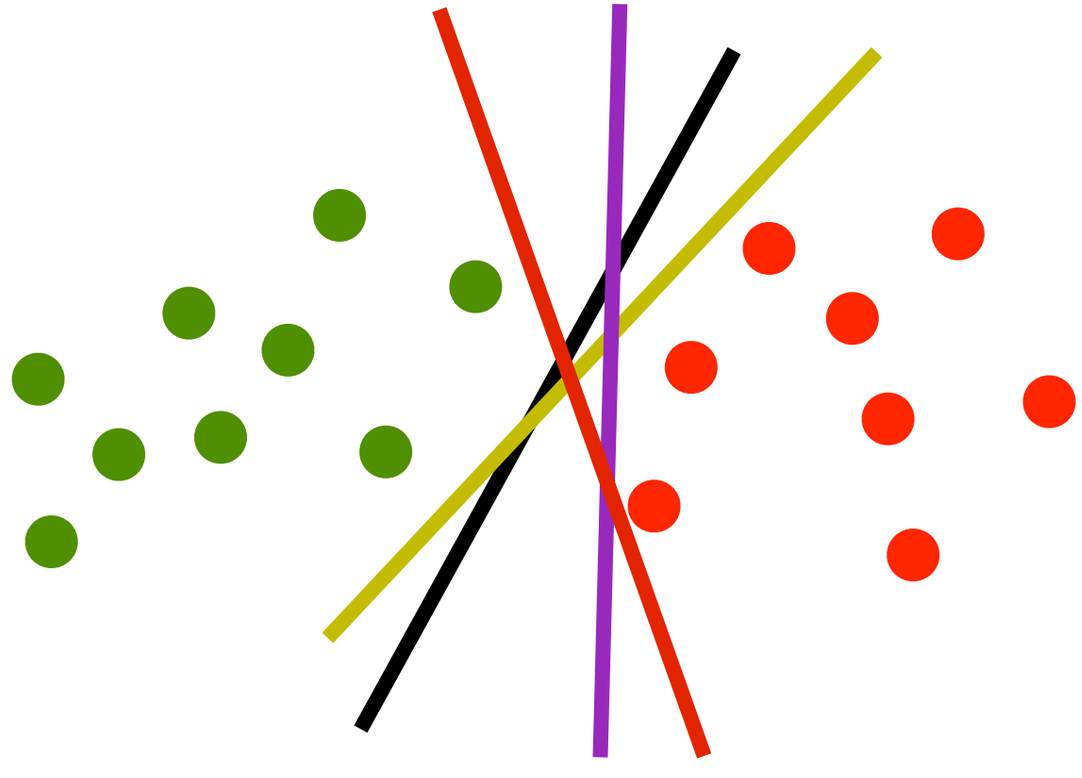
→ We rewrite the functional using slack variables

A more classical formulation (linear case)

$$\omega^* = \min_{\omega \in \mathbb{R}^d, \xi_i \geq 0} \|\omega\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad y_i \omega^T x_i \geq 1 - \xi_i, \quad \forall i \in \{1, \ldots, n\}$$

In general there are many solutions!!!



Which one do you select?

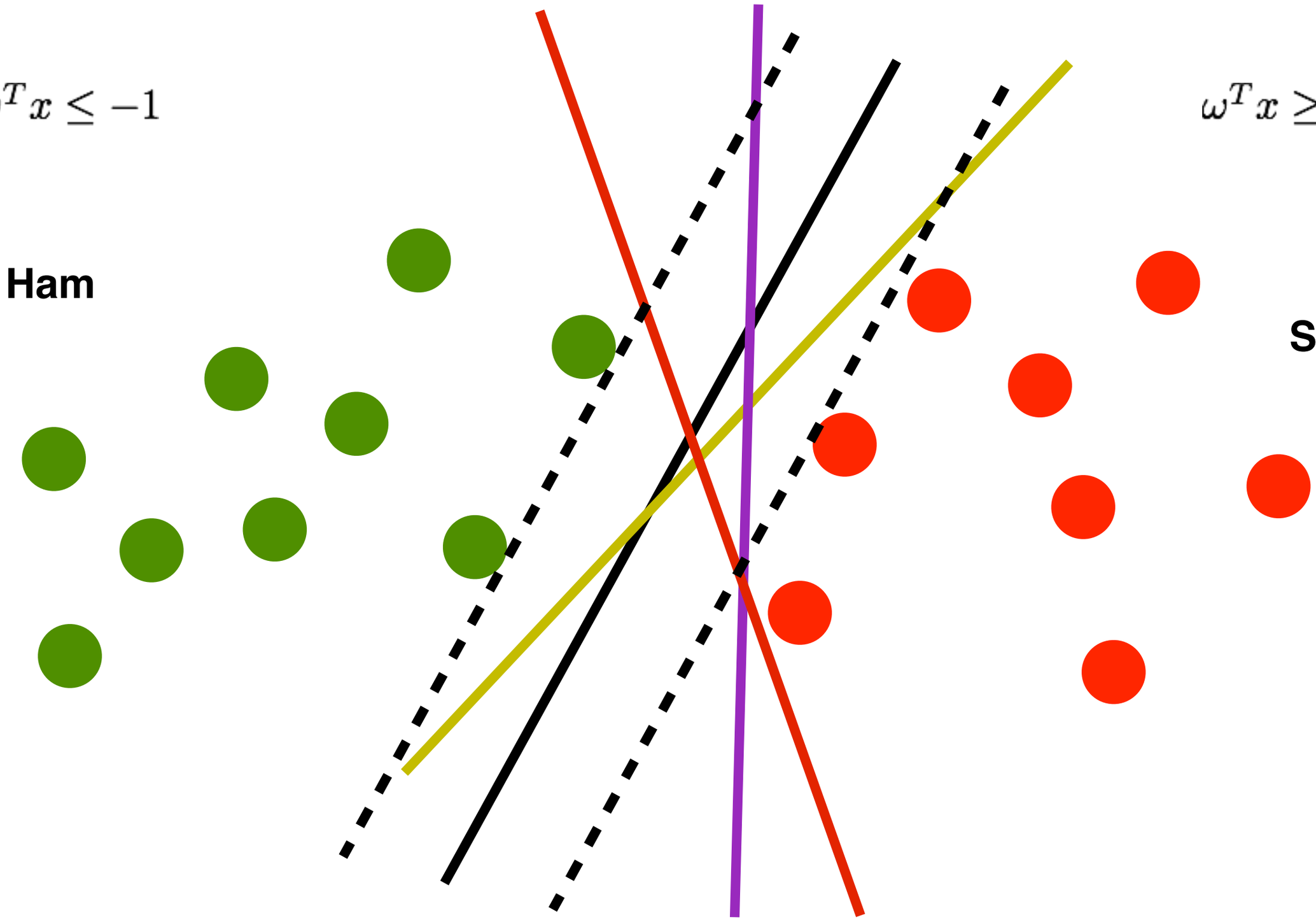
Linear Separator

$\omega^T x \leq -1$

$\omega^T x \geq 1$

Ham

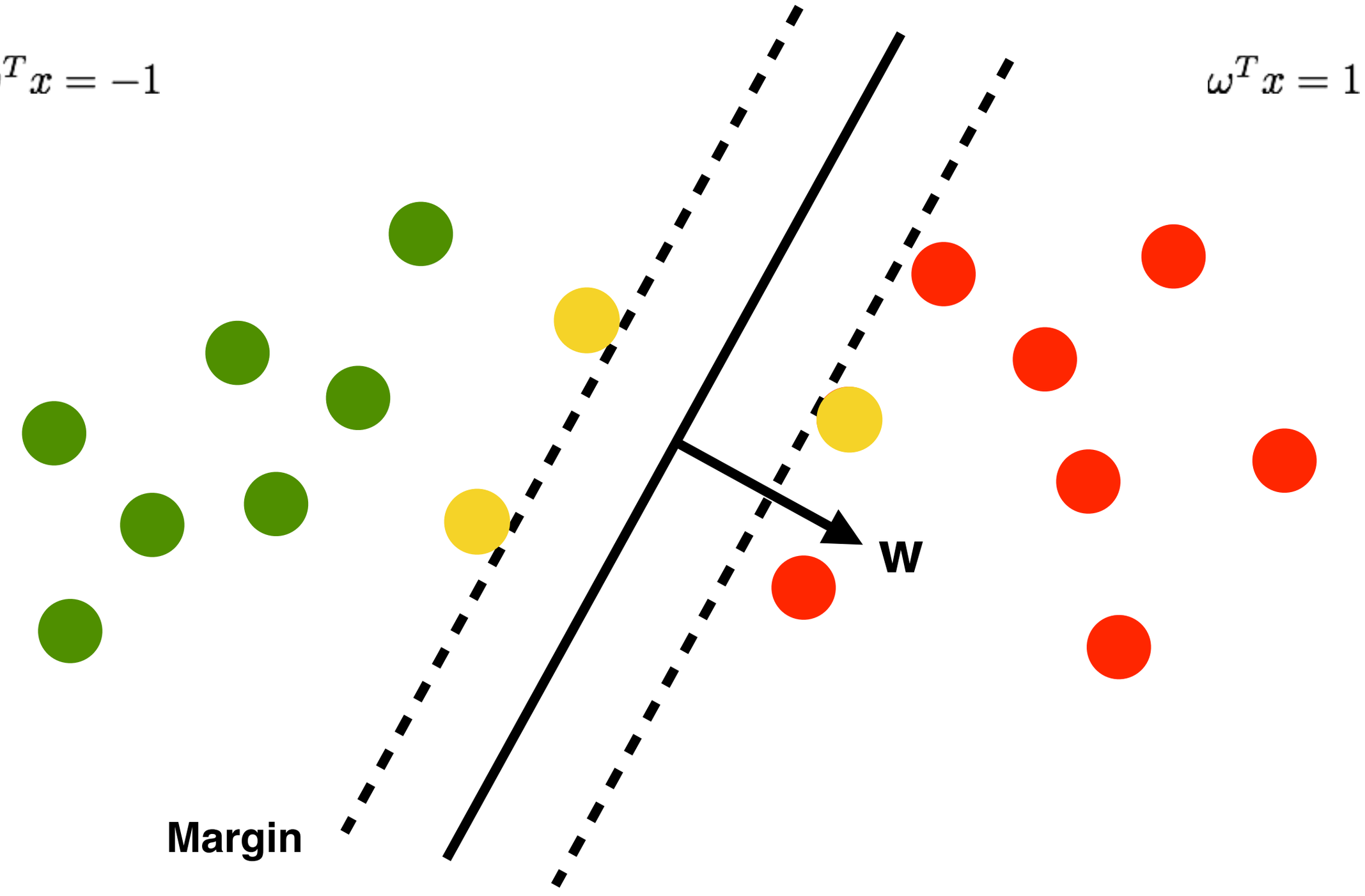
Spam



Linear function

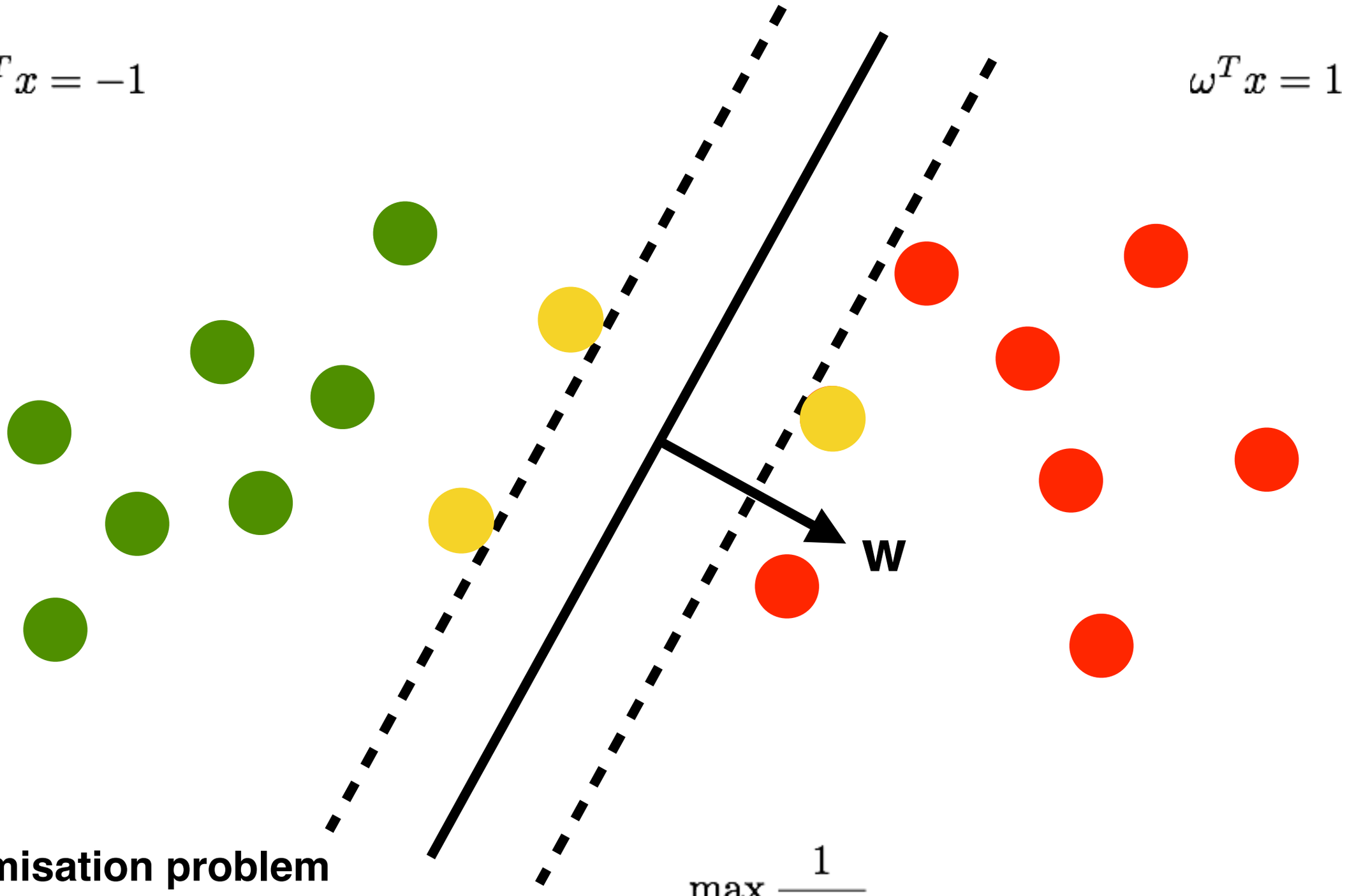
$$f(x) = \omega^T x$$

Large Margin Classifier



$$\frac{\omega^T(x_+ - x_-)}{2\|\omega\|} = \frac{1}{2\|\omega\|}\left(\omega^T x_+ - \omega^T x_-\right) = \frac{1}{\|\omega\|}$$

Large Margin Classifier

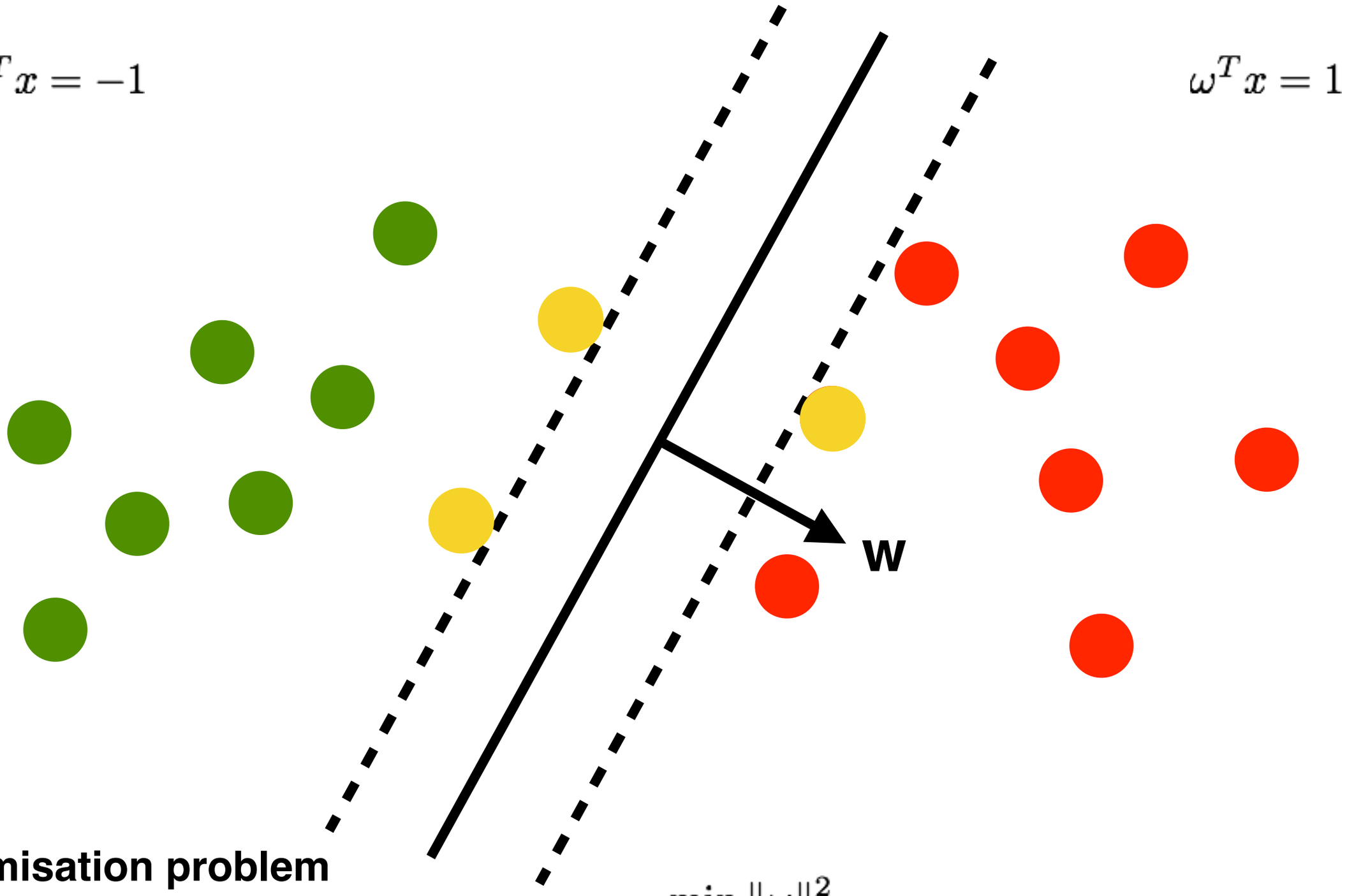


Optimisation problem

$$\max_{\omega} \frac{1}{\|\omega\|^2}$$

$$\text{subject to } y_i \omega^T x_i \geq 1, \quad \forall i \in \{1, \ldots, n\}$$

Large Margin Classifier



Optimisation problem

$$\min_{\omega} \|\omega\|^2$$

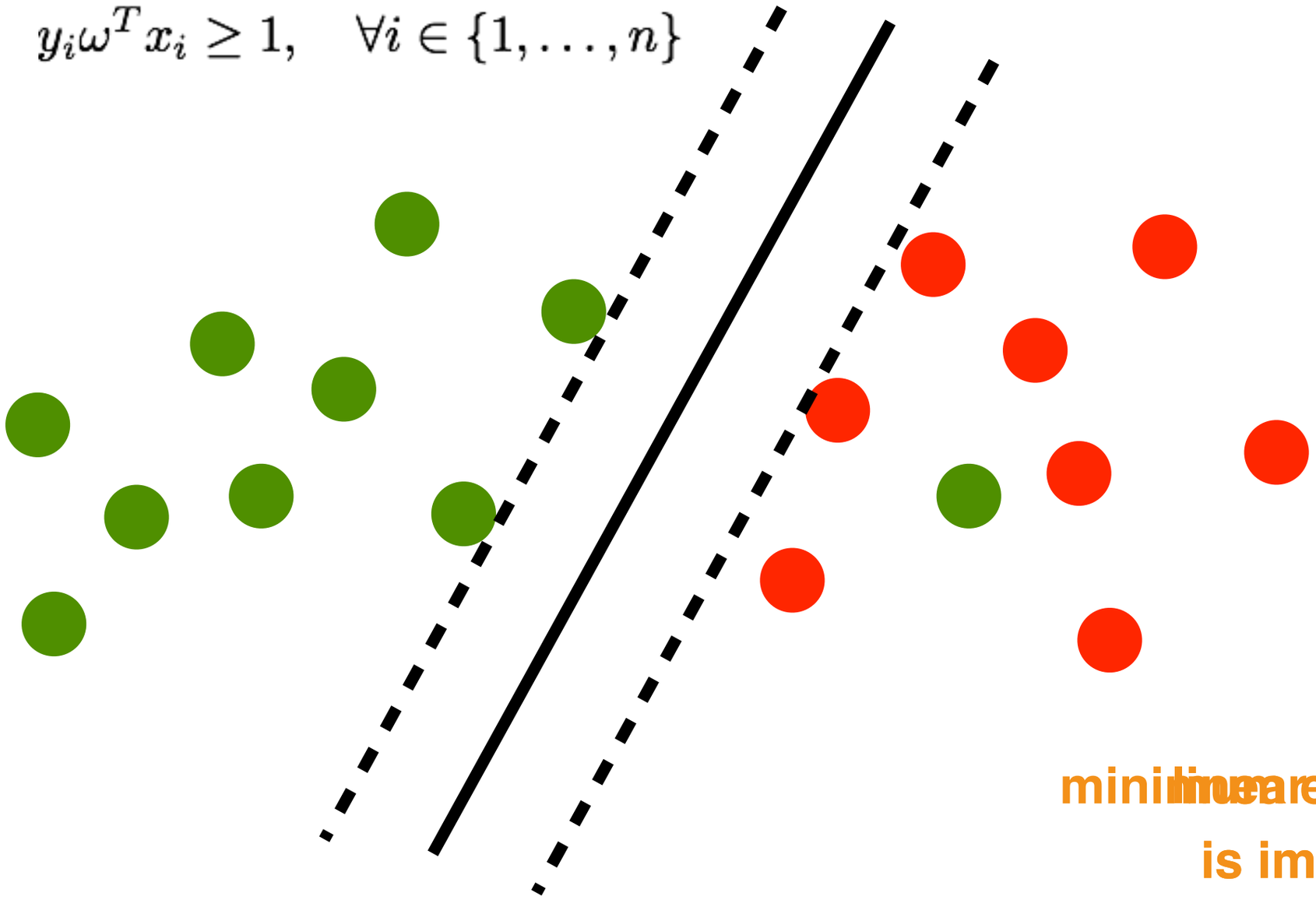
$$\text{subject to } y_i \omega^T x_i \geq 1, \quad \forall i \in \{1, \ldots, n\}$$

What if the problem is not separable?

We relax the constraints and penalise the relaxation

$$\min_{\omega} \|\omega\|^2$$

$$y_i \omega^T x_i \geq 1, \quad \forall i \in \{1, \ldots, n\}$$

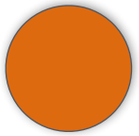


minimum error separator is impossible

Theorem (Minsky & Papert)

Finding the minimum error separating hyperplane is NP hard

What if the problem is not separable?



We obtain a convex optimisation problem

$$\omega^* = \min_{\omega \in \mathbb{R}^d, \xi_i \geq 0} \|\omega\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad y_i \omega^T x_i \geq 1 - \xi_i, \quad \forall i \in \{1, \ldots, n\}$$

where C is a penalisation parameter for the average error $\frac{1}{n} \sum_{i=1}^{n} \xi_i$

Problem is always feasible!!!

Intermezzo: Convex Programmes

- Primal optimisation problem

$$\underset{x}{\text{minimize}}\, f(x) \text{ subject to } c_i(x) \leq 0$$

- Lagrange function

$$L(x, \alpha) = f(x) + \sum_i \alpha_i c_i(x)$$

- First order optimality conditions in x

$$\partial_x L(x, \alpha) = \partial_x f(x) + \sum_i \alpha_i \partial_x c_i(x) = 0$$

- Solve for x and plug it back into L

$$\underset{\alpha}{\text{maximize}}\, L(x(\alpha), \alpha)$$

(keep explicit constraints)

Dual Formulation

Applying the above-mentioned steps, it can be shown that the solution of the SVM problem is of the form

$$\omega = \sum_{i=1}^{n} \alpha_i y_i x_i$$

where α_i are given by the solution of the following quadratic programming problem:

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j \qquad i = 1, \ldots, n$$
$$\text{s.t. } \alpha_i \geq 0$$

- The solution requires the estimate of *n* rather than *D* coefficients.
- Coefficients α_i are often sparse. The input points associated with non-zero coefficients are called *support vectors*.
- If problem is small enough (1000s of variables) one can use off-the-shelf quadratic solver (CVXOPT, LOQO).

End of sub-Part II

We considered the regularised empirical risk minimisation with different loss function, leading to different regularisation methods.

Regularised Empirical Risk Minimisation:

$$\omega^* = \min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} V(y, f(x)) + \lambda \|\omega\|^2$$

Examples of Regularisation Networks:

- Least Squares (Square loss): $V(y,t) = (y-t)^2$
- Logistic Regression (Logistic loss): $V(y,t) = \log(1 + e^{-yt})$
- Maximum Margin Classifier (Hinge loss): $V(y,t) = |1 - yt|_+$

Logistic Regression - Large Margin Classifier - SVM

Part IIIa. Dimensionality Reduction

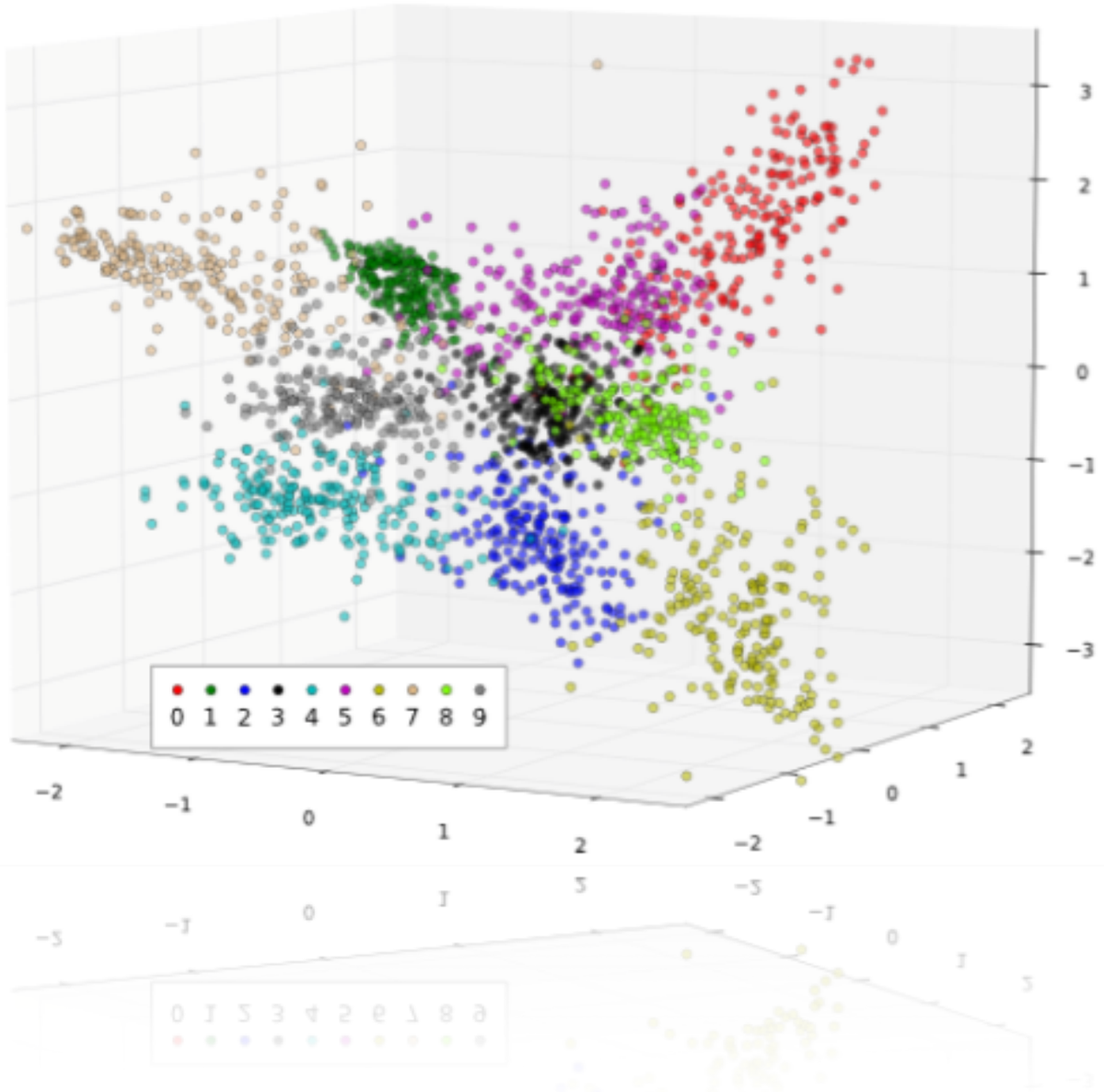
GOAL:

To introduce methods that allow to reduce data dimensionality in absence of labels, namely **unsupervised learning**

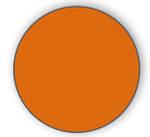
Dimensionality Reduction for Data Visualisation

In many practical applications it is of interest to reduce the dimensionality of the data:

- data visualisation
- data exploration: for investigating the ‘effective’ dimensionality of the data



Dimensionality Reduction



The problem of dimensionality reduction can be seen as the problem of defining a map

$$M : X = \mathbb{R}^D \to \mathbb{R}^k, \quad k \ll D$$

according to some *suitable criterion.*

We will follow **data reconstruction** as our guiding principle.

Principle Component Analysis

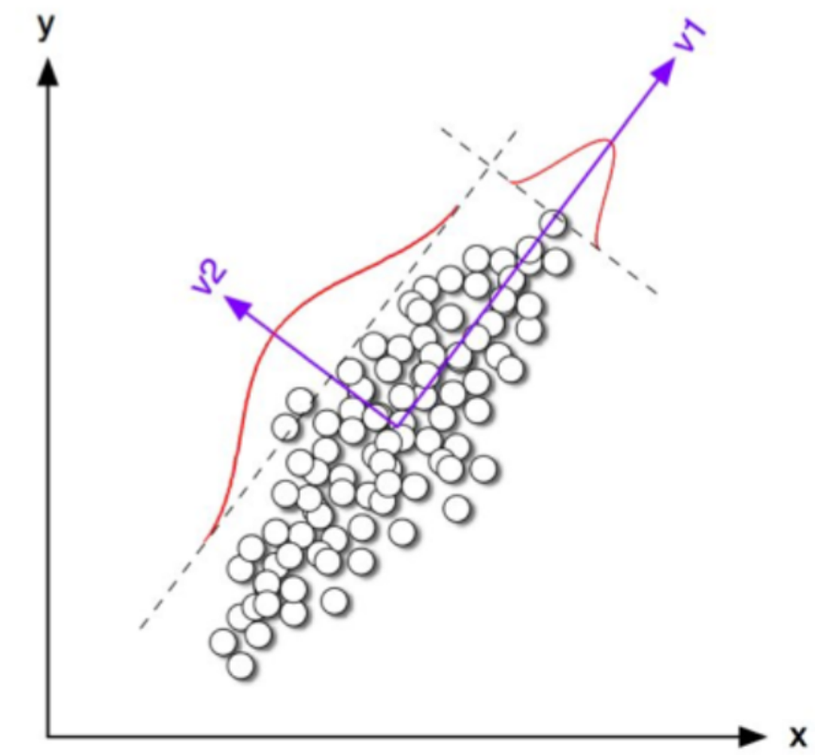
PCA is arguably the most popular dimensionality reduction procedure

It is a data driven procedure that given an **unsupervised sample**

$$S = (x_1, \ldots, x_n)$$

derive a dimensionality reduction defined by a linear map *M.*

PCA can be derived from several prospective and here we give a **geometric** derivation.

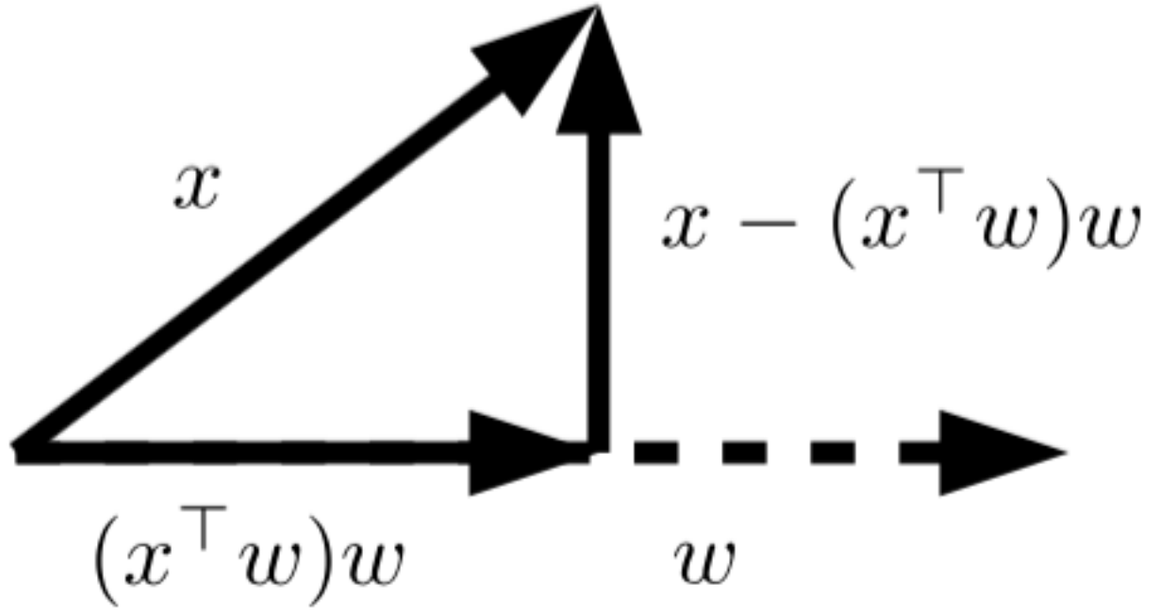


Principle Component Analysis

Recall that if

$$\omega \in \mathbb{R}^D, \quad \|\omega\| = 1,$$

then $(\omega^T x)\omega$ is the **orthogonal projection** of x on ω



Dimensionality Reduction

Consider at first k =1. The associated **reconstruction error** is

$$\|x - (\omega^T x)\omega\|^2$$

That is how much we lose by projecting x along the direction ω

Problem: Find the direction p allowing the best reconstruction of the training set

Consider the **empirical reconstruction** minimisation problem

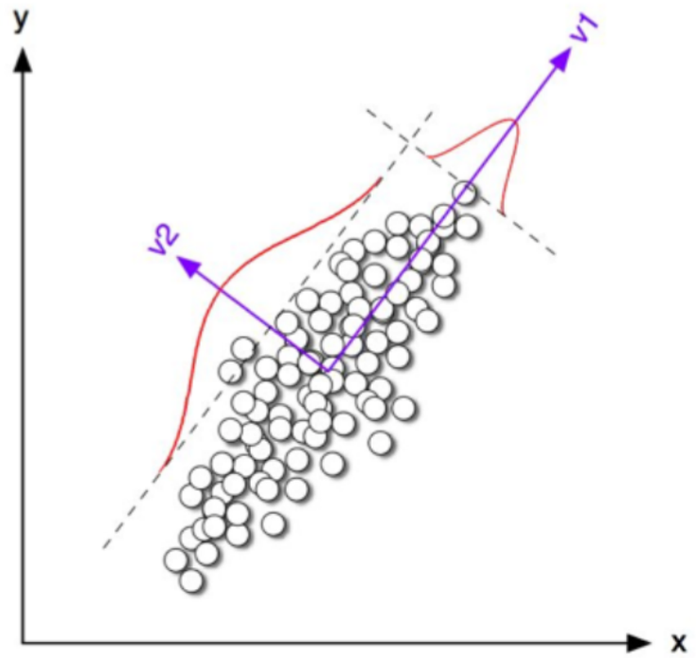
$$\min_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - (\omega^T x_i)\omega\|^2$$

$$\omega^T \omega = 1$$

The solution p to the above problem is called the **first principal component** of the data

Computations?

Statistics?



PCA: An Equivalent Formulation

$$\min_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - (\omega^T x_i)\omega\|^2$$

Statistics?

$$\|x_i - (\omega^T x_i)\omega\|^2 = \|x_i\|^2 - (\omega^T x_i)^2$$

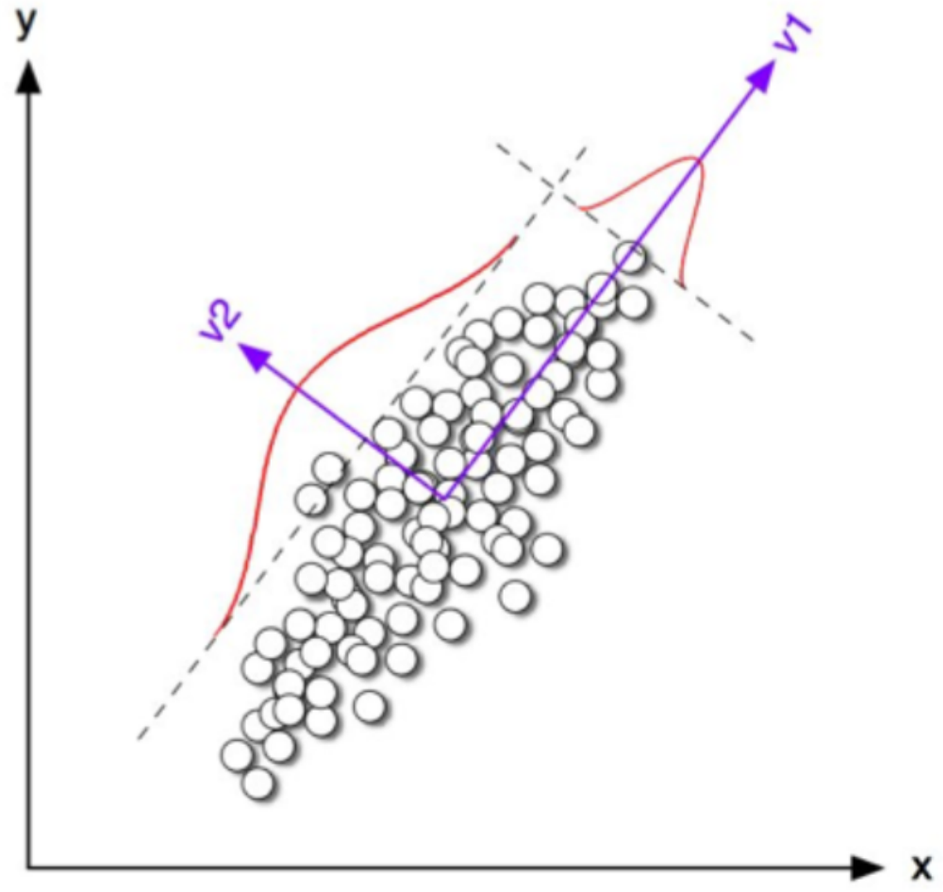
centered data →

$$\max_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} (\omega^T x_i)^2$$

Variance of x in the direction ω

not centered data →

$$\max_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} (\omega^T (x_i - \bar{x}))^2$$



PCA as an Eigenproblem

$$\min_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - (\omega^T x_i)\omega\|^2$$

Computations?

$$\max_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} (\omega^T x_i)^2 \quad \longleftrightarrow \quad \max_{\omega \in \mathbb{S}^{D-1}} \omega^T C_n \omega, \quad C_n = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

ω_1 max eigenvector of C_n

$$\frac{1}{n} \sum_{i=1}^{n} (\omega^T x_i)^2 = \frac{1}{n} \sum_{i=1}^{n} \omega^T x_i \omega^T x_i = \frac{1}{n} \sum_{i=1}^{n} \omega^T x_i x_i^T \omega = \omega^T \left(\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right) \omega$$

PCA as an Eigenproblem

$$\min_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - (\omega^T x_i)\omega\|^2$$

Computations?

$$\max_{\omega \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} (\omega^T x_i)^2 \quad \longleftrightarrow \quad \max_{\omega \in \mathbb{S}^{D-1}} \omega^T C_n \omega, \quad C_n = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

ω_1 max eigenvector of C_n

Computing the first principal component of the data reduces to computing the biggest eigenvalue of the covariance and the corresponding eigenvector.

$$C_n u = \lambda u, \qquad C_n = \frac{1}{n} \sum_{i=1}^{n} X_n^T X_n$$

Dimensionality Reduction

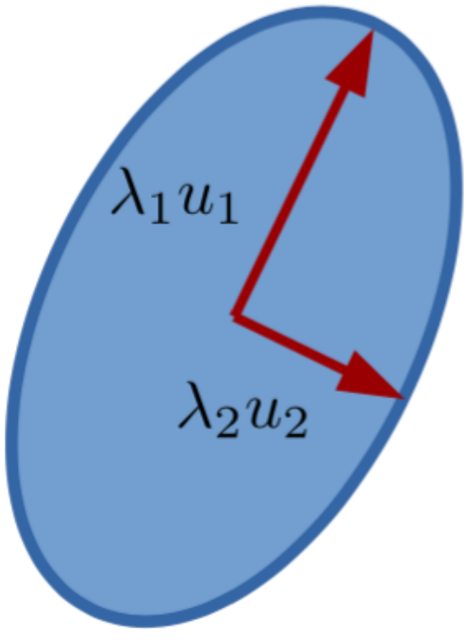
$$M : X = \mathbb{R}^D \to \mathbb{R}^k, \quad k \ll D$$

What about *k* =2??? **Idea:** simply iterate previous reasoning

….

$$\max_{\omega \in \mathbb{S}^{D-1}; \omega \perp \omega_1} \omega^T C_n \omega, \qquad C_n = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

ω_2 second eigenvector of C_n



The reasoning generalises to more than two components…

Remarks

Computational complexity: $O(kD^2)$

(complexity of forming C_n is $O(nD^2)$)

If we have *n* points in *D* dimensions and $n \ll D$ can we compute PCA in less than $O(nD^2)$?

The dimensionality reduction induced by PCA is a linear projection. Can we generalise PCA to non linear dimensionality reduction?

Beyond Linear Dimensionality Reduction?

By considering PCA we are implicitly assuming the data to lie on a linear

subspace....

...it is easy to think of situations where this assumption might violated.

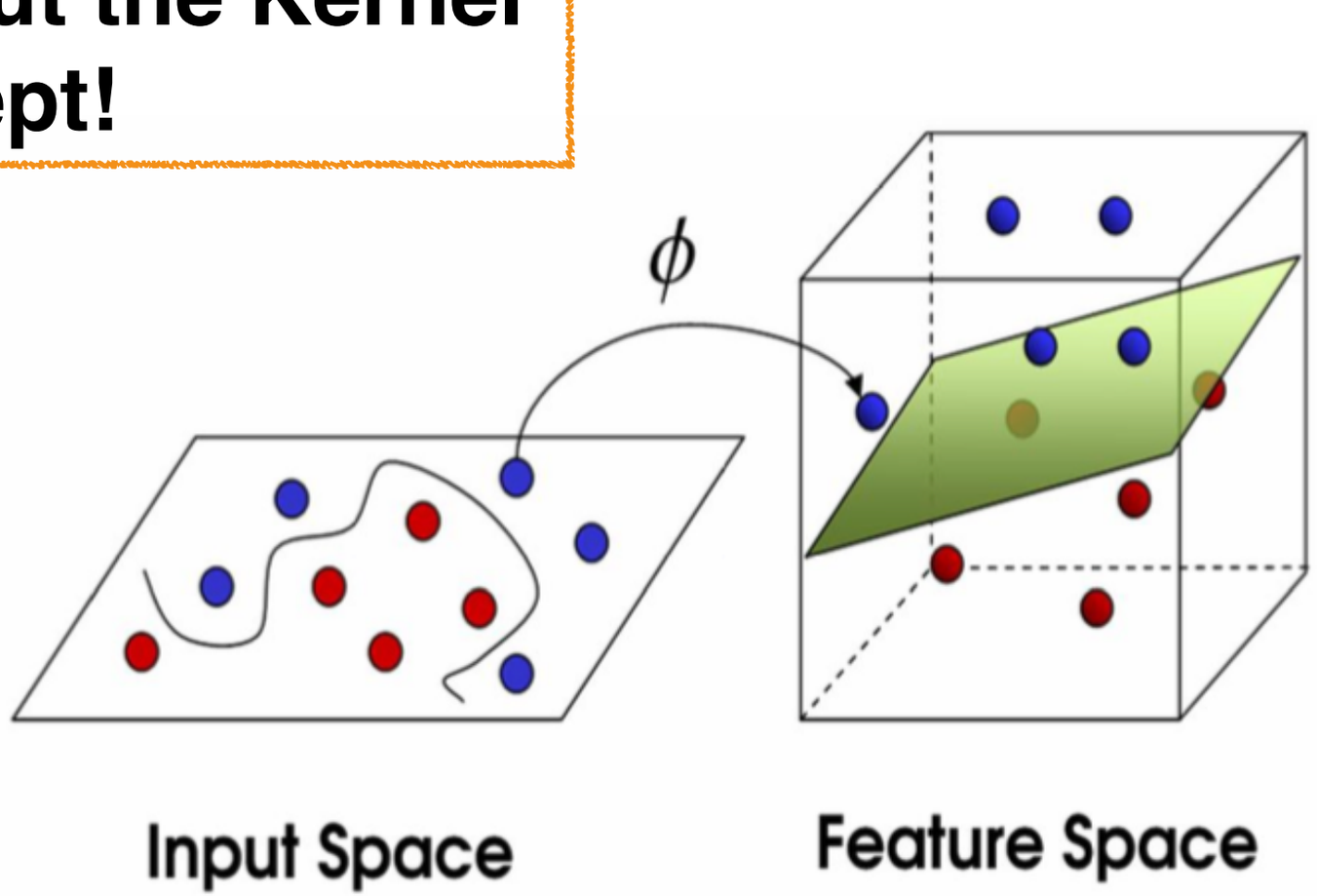
Can we use kernels to obtain non linear generalisation of PCA?

PCA and Feature Maps

What if consider a non linear feature-map $\Phi : X \rightarrow F$ before performing PCA?

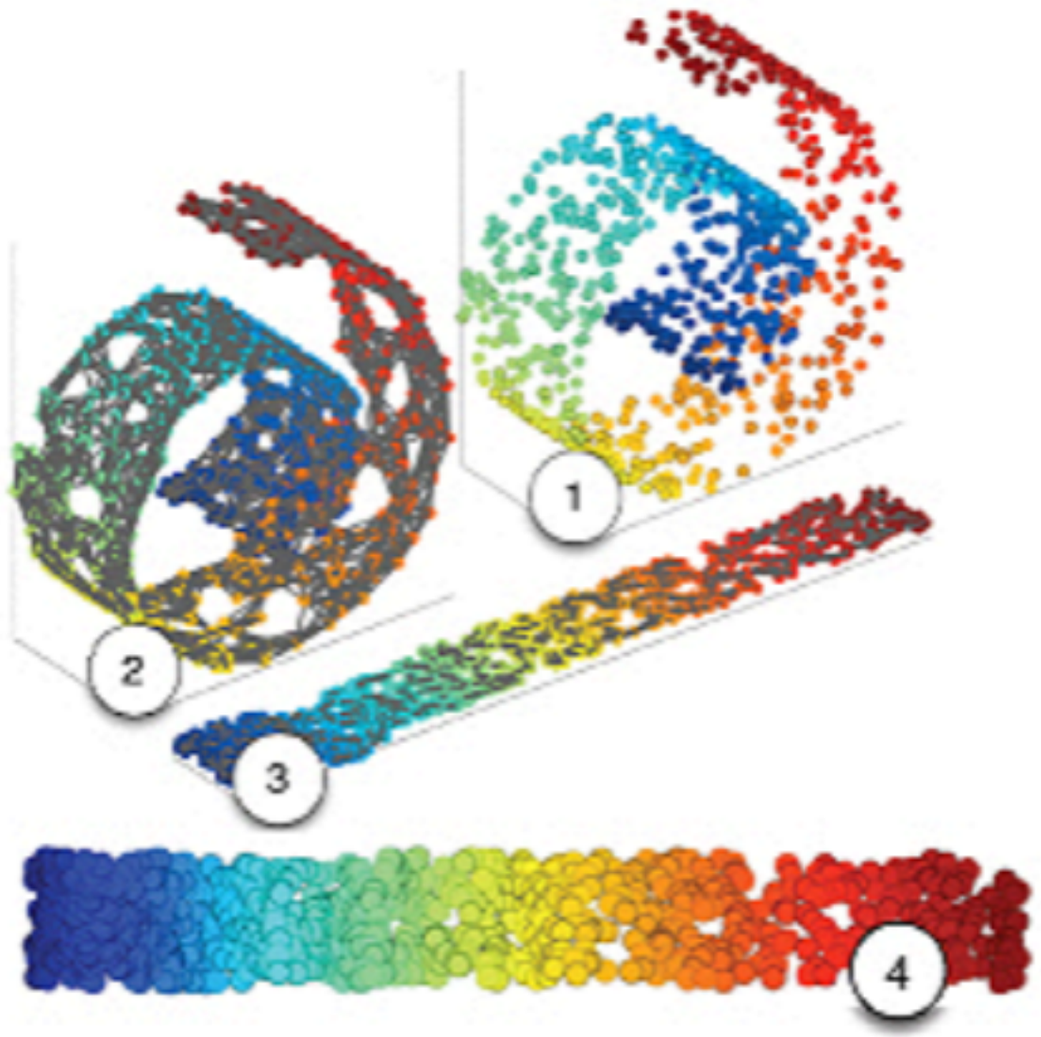
Remember about the Kernel concept!

$$K(x, x_i) = \Phi(x)^T \Phi(x_i)$$



Things I won't tell you about this time….

- **Random maps:** Johnson-Linderstrauss Lemma
- **Non Linear Maps:** Laplacian / Diffusion maps



End of Part IIIa

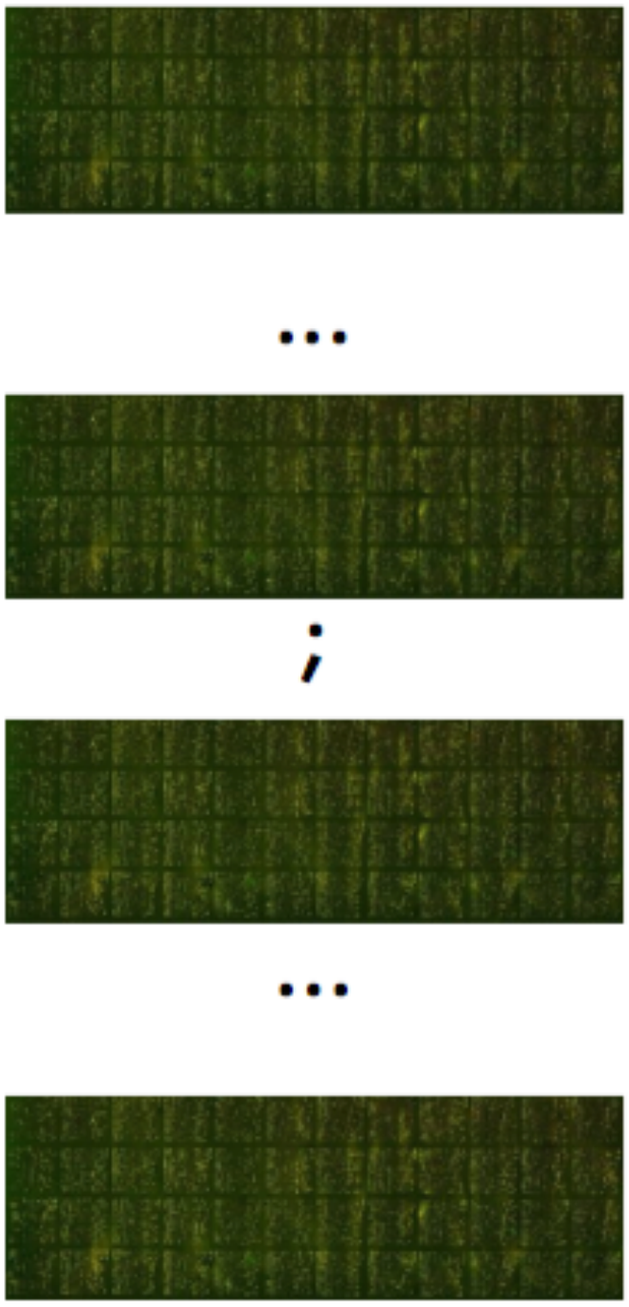
PCA - Dimensionality Reduction -

Eigenvectors - Principal Components

Part IIIb. Variable Selection

GOAL:

To introduce methods that allow to learn *interpretable* models from data



...

;

...

***n* patients *p* gene expression measurements**

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$$f(x) = \omega^T x = \sum_{j=1}^{p} x^j \omega^j$$

Which variables are important for prediction?

or

Torture the data until they confess

We look at this question from the perspective of **variable selection**

Linear Models

Consider a linear model

$$f(x) = \omega^T x = \sum_{j=1}^{p} x^j \omega^j$$

- The components x^j of an input can be seen as **measurements** (pixel values, gene expression, dictionary words count…)
- Given data, the goal of variable selection is to detect which are **variables important for prediction**

Key Assumption: the best possible prediction rule is **sparse**, that is only few of the coefficients are non zero

Notation

We need some notation

- X_n be the n by D data matrix
- $X^j \in \mathbb{R}^n, \quad j = 1, \ldots, D$ its columns
- $Y_n \in \mathbb{R}^n$ the output vector

Estimating a linear model corresponds to solving a linear system

$$X_n \omega = Y_n$$

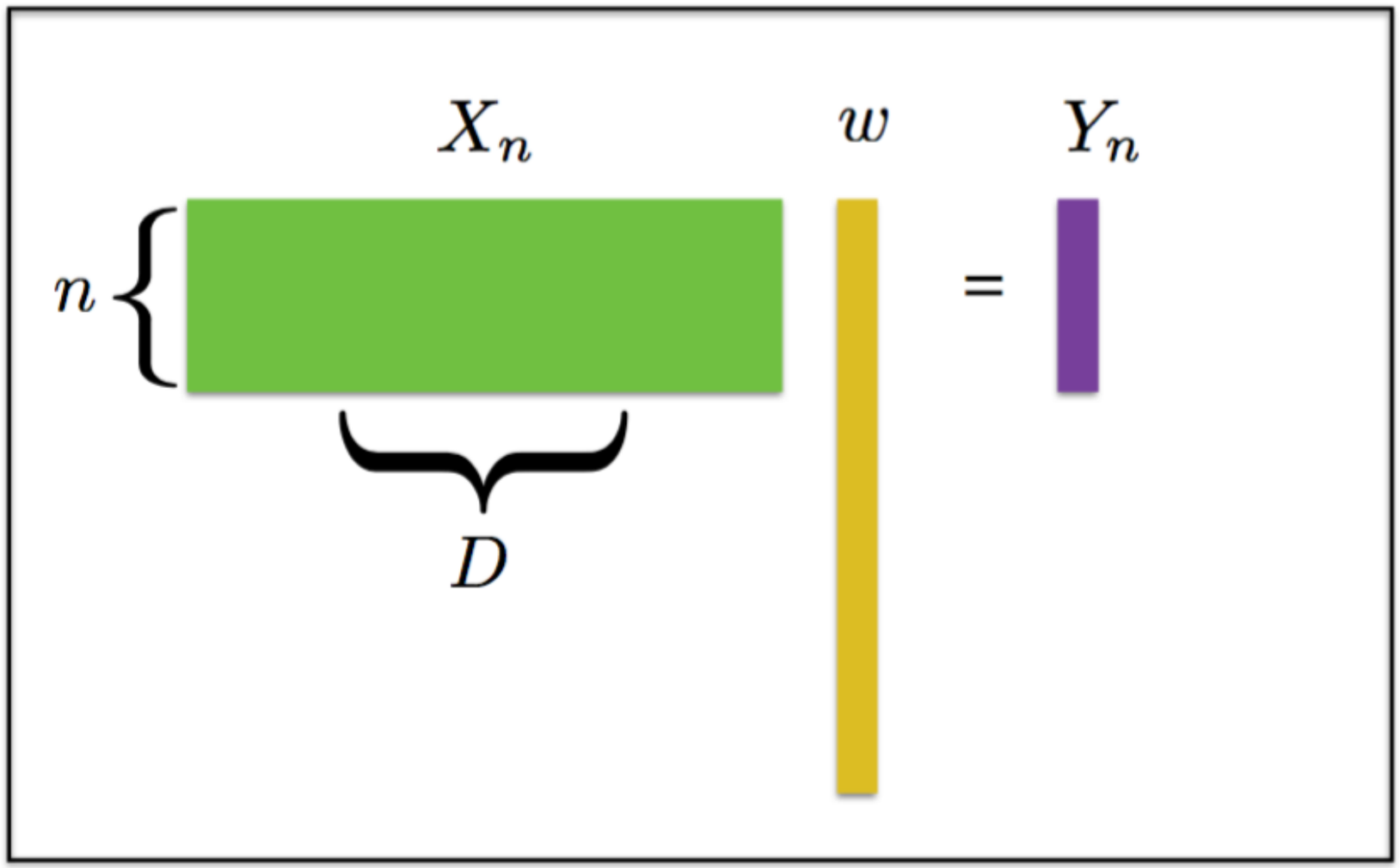
- Classically $n \gg D$ **low dimension/overdetermined system**
- Lately $n \ll D$ **high dimensional/underdetermined system**

Buzzwords: compressed sensing, high-dimensional statistics . . .

High-Dimensional Statistics

Estimating a linear model corresponds to solving a linear system

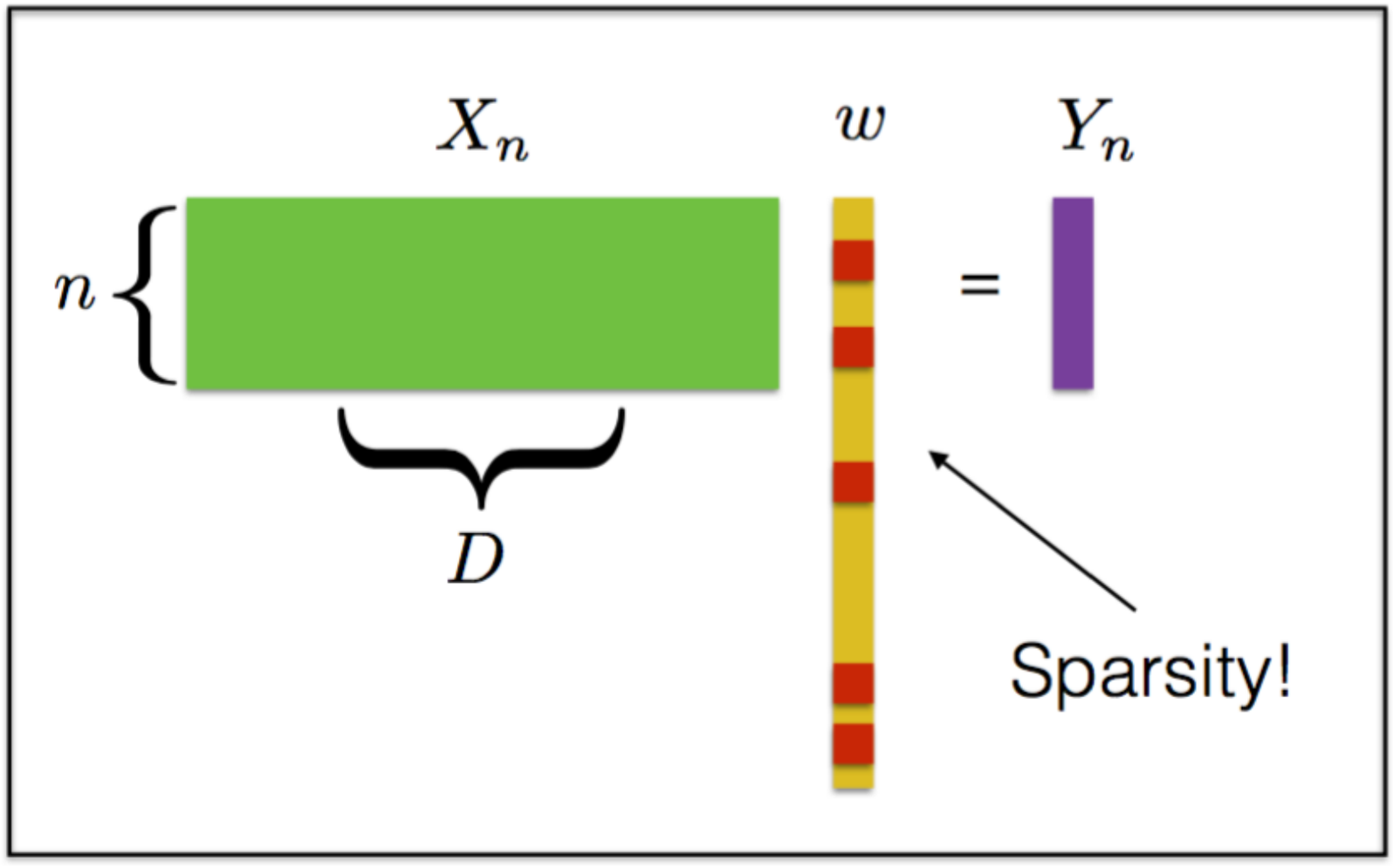
$$X_n \omega = Y_n$$



High-Dimensional Statistics

Estimating a linear model corresponds to solving a linear system

$$X_n \omega = Y_n$$



Brute Force Approach

…check all individual variables, then all couple, triplets.....

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|\omega\|_0$$

where $\|\omega\|_0 = |\{j : \omega^j \neq 0\}|$ counts non zero components in the vector.

→ **The computational complexity is combinatorial!!!**

We consider two possible approximate approaches:

- Greedy methods
- Convex relaxation

Brute Force Approach

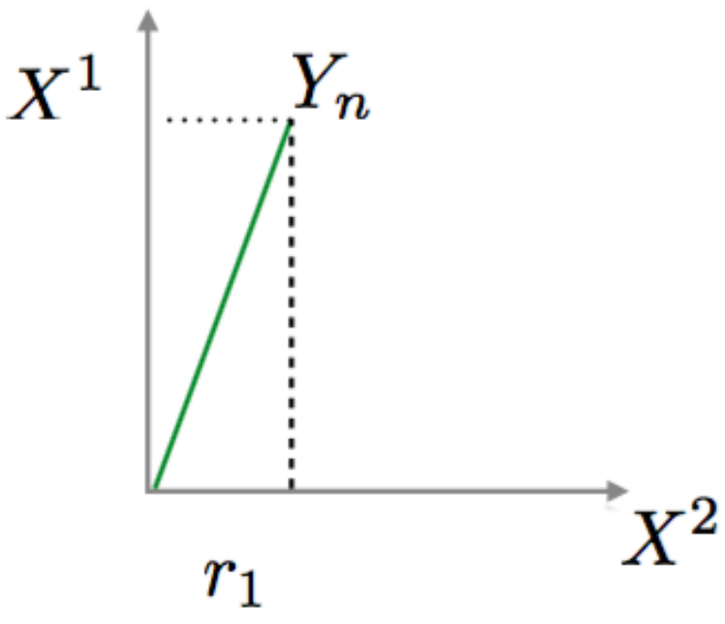
…check all individual variables, then all couple, triplets.....

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|\omega\|_0$$

where $\|\omega\|_0 = |\{j : \omega^j \neq 0\}|$ counts non zero components in the vector.

→ **The computational complexity is combinatorial!!!**

Greedy Approaches (one possibility to solve the problem)



1. initialise the residual, the coefficient vector, and the index set
2. find the variable most correlated with the residual
3. update the index set to include the index of such variable
4. update/compute coefficient vector,
5. update residual.

The simplest such procedure is called matching pursuit (MP) in signal processing

Matching Pursuit

Mallat, Zhang, 1993

Let r, ω, I denote the residual, the coefficient vector, an index set.

$$r_0 = Y_n, \quad , w_0 = 0, \quad I_0 = \emptyset.$$ **Initialisation**

Matching Pursuit

Mallat, Zhang, 1993

Let r, ω, I denote the residual, the coefficient vector, an index set.

$$r_0 = Y_n, \quad , w_0 = 0, \quad I_0 = \emptyset.$$ **Initialisation**

for $i = 1, \ldots, T-1$

$$k = \arg\max_{j=1,\ldots,D} a_j, \quad a_j = \frac{(r_{i-1}^T X^j)^2}{\|X^j\|^2}, \circledast$$ **The variable most correlated with residual**

Such a selection rule has two interpretations:

- We select the variable with **larger** projection on the output, or equivalently.
- We select the variable such that the corresponding column best explains the the output vector in a **least squares sense.**

$$\circledast \quad v^j = \frac{r_{i-1}^T X^j}{\|X^j\|^2} = \arg\min_{v \in \mathbb{R}} \|r_{i-1} - X^j v\|^2, \quad \|r_{i-1} - X^j v^j\|^2 = \|r_{i-1}\|^2 - a_j$$

Matching Pursuit

Mallat, Zhang, 1993

Let r, ω, I denote the residual, the coefficient vector, an index set.

$$r_0 = Y_n, \quad , w_0 = 0, \quad I_0 = \emptyset.$$ **Initialisation**

for $i = 1, \ldots, T-1$

$$k = \arg\max_{j=1,\ldots,D} a_j, \quad a_j = \frac{(r_{i-1}^T X^j)^2}{\|X^j\|^2}, \circledast$$ **The variable most correlated with residual**

$$I_i = I_{i-1} \cup \{\hat{k}\}$$

$$w_i = w_{i-1} + w_k, \quad w_k k = v_k e_k$$ **Update of the index set and the coefficient vector**

$$\circledast \; v^j = \frac{r_{i-1}^T X^j}{\|X^j\|^2} = \arg\min_{v \in \mathbb{R}} \|r_{i-1} - X^j v\|^2, \quad \|r_{i-1} - X^j v^j\|^2 = \|r_{i-1}\|^2 - a_j$$

Matching Pursuit

Mallat, Zhang, 1993

Let r, ω, I denote the residual, the coefficient vector, an index set.

$r_0 = Y_n, \quad , w_0 = 0, \quad I_0 = \emptyset.$ Initialisation

for i

ed with

$r_i = r_{i-1} - X w^*.$

Residual update

end

Update of the index set and the coefficient vector

$\circledast \quad v^j = \frac{r_{i-1}^T X^j}{\|X^j\|^2} = \arg\min_{v \in \mathbb{R}} \|r_{i-1} - X^j v\|^2, \quad \|r_{i-1} - X^j v^j\|^2 = \|r_{i-1}\|^2 - a_j$

Theoretical guarantees

If

- the solution is **sparse**, and
- the data matrix has columns '**not too correlated**'

Orthogonal Matching Pursuit can be shown to **recover** with high probability the **right vector of coefficients**

Basis Pursuit / Lasso

Donoho etal. 1995; Tibshirani 1996

Another popular approach to find sparse solutions is based on a **convex relaxation**

$$\|\omega\|_1 = \sum_{j=1}^{D} |\omega^j|$$

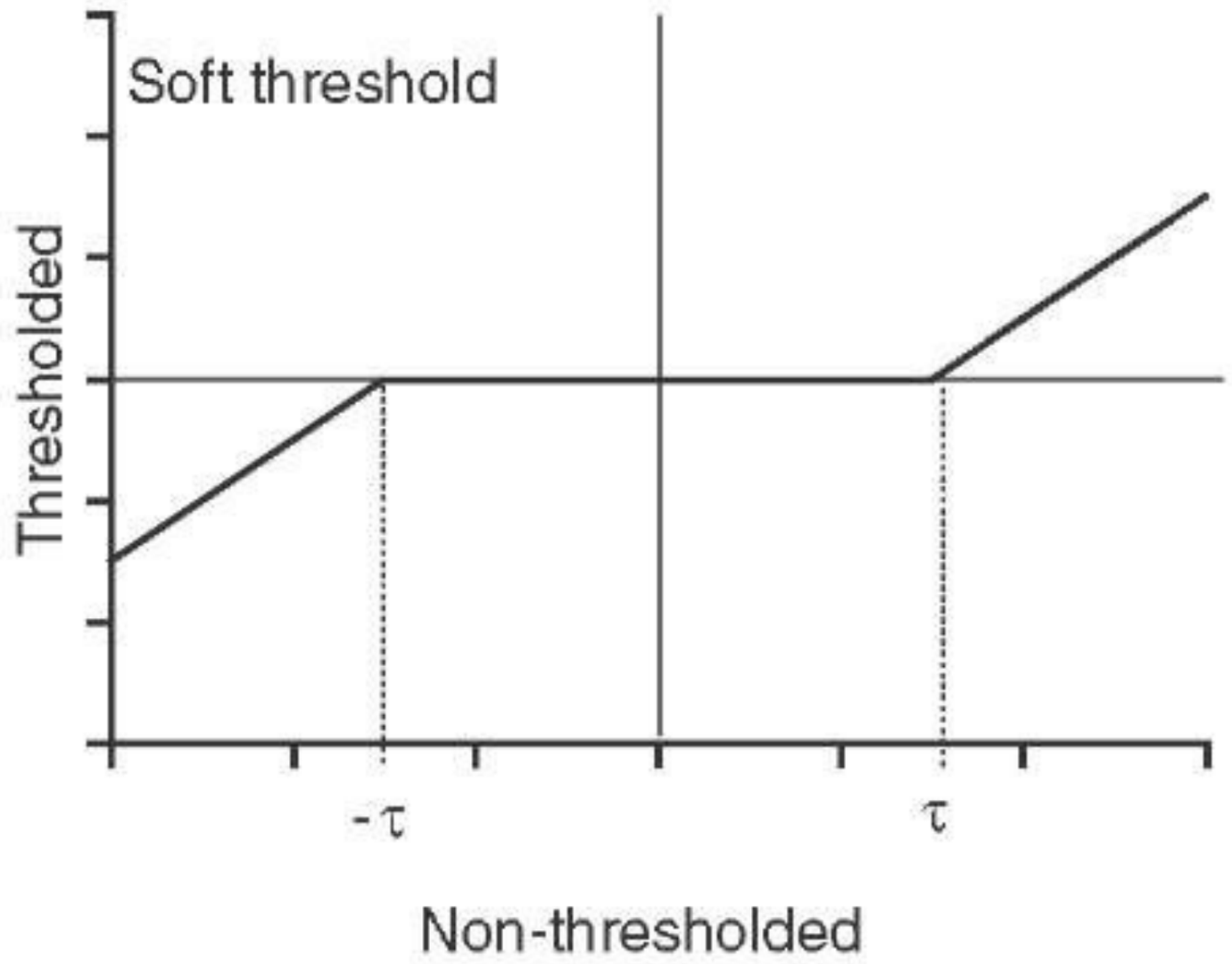
$$\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|\omega\|_0$$

Problem is now **convex** and can be solved using convex optimisation, in particular so called *proximal methods*

Iterative Soft Thresholding

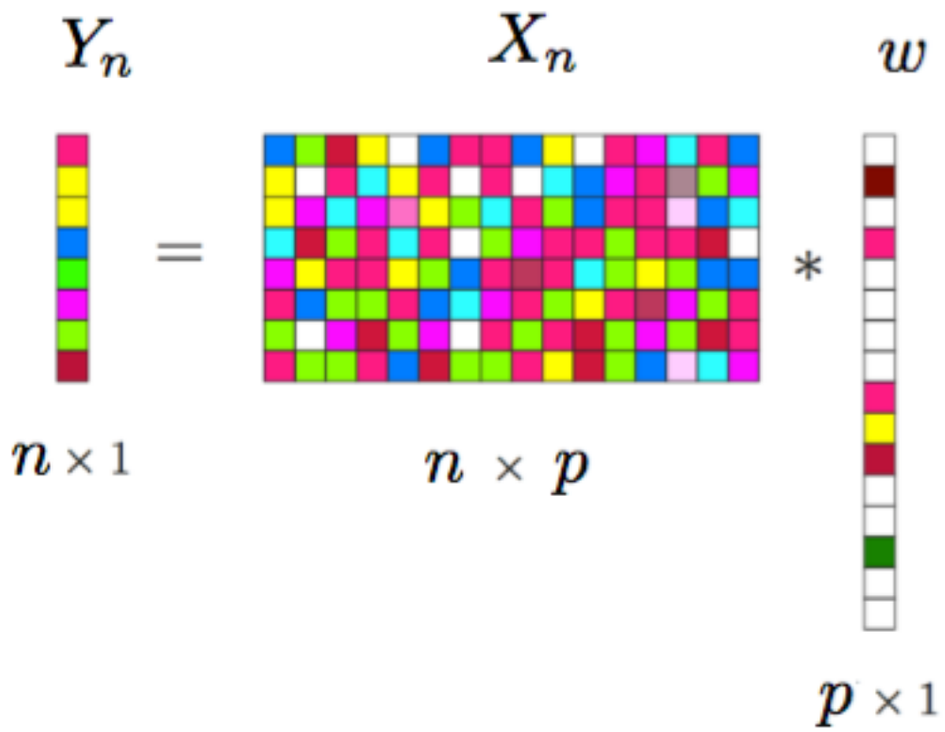
A simple yet powerful procedure to compute a solution is based on the so-called **iterative soft thresholding algorithm (ISTA):**

$$\omega_0 = 0, \qquad \omega_i = S_{\lambda,\gamma}(\omega_{i-1} - \tfrac{2\gamma}{n} X_n^T (Y_n - X_n \omega_{i-1})), \quad i = 1, \ldots, T_{\max}$$



Things I won't tell you about this time….

- Solving underdetermined systems
- Sampling theory
- Compressed Sensing
- Structured Sparsity
- From vector to matrices- from sparsity to low rank



Mini-tutorial by Jean-Luc Starck (CosmoStat)

Sparse Representations and their Applications

September 02, 2016

End of Part IIIb

Interpretability - Sparsity -

Greedy and Convex Relaxation Approaches

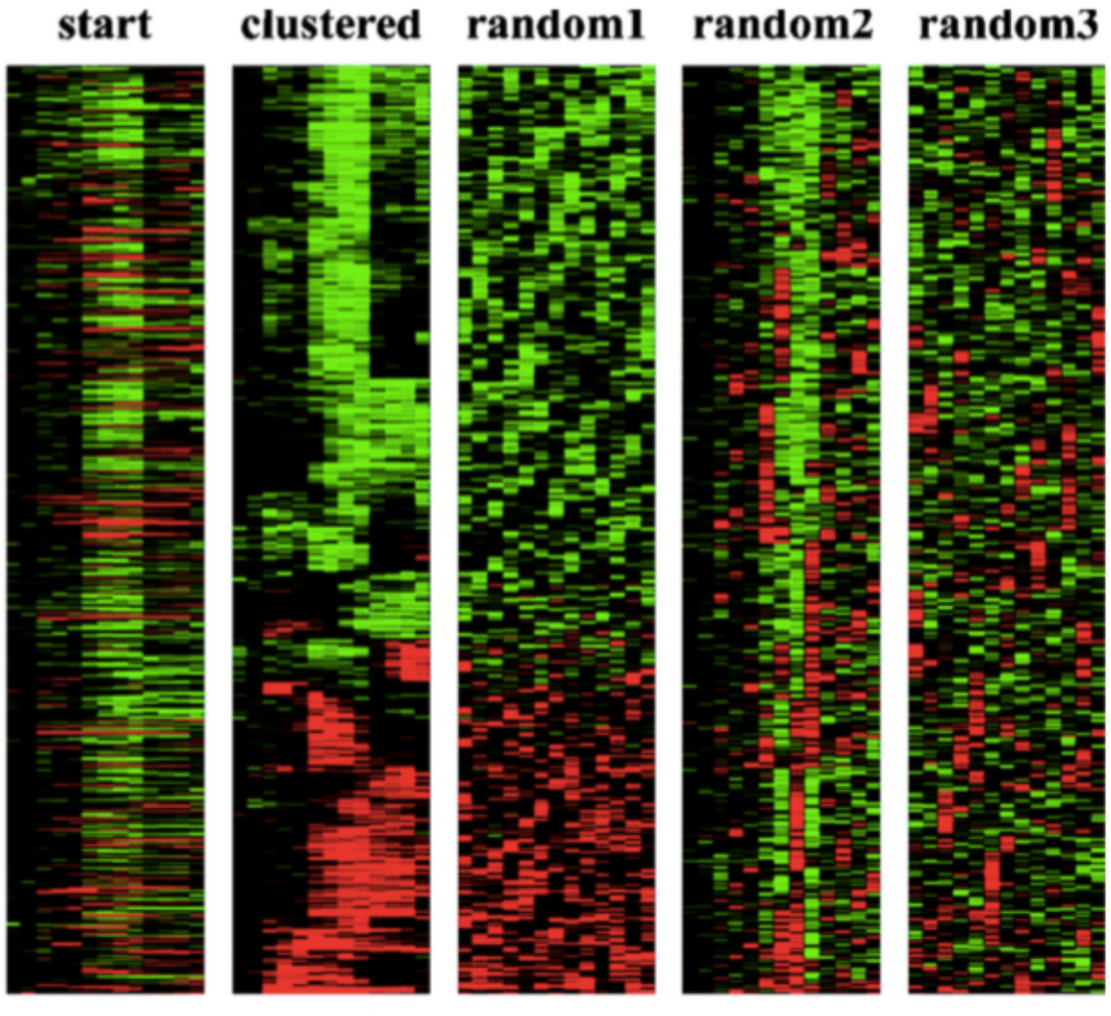
Part IV. Clustering

GOAL:

Introduce methods for partitioning of the observed unlabelled data.

Examples and Algorithms

Mixture Models



Michael B. Eisen et al. PNAS 1998;95:14863-14868

Combinatorial Algorithms

Combinatorial Algorithms

- We assume some knowledge on the number of clusters K ≤ n.
  Goal: associate a cluster label $k = \{1, \ldots, K\}$ with each datum, by defining an encoder *C* s.t.

$$k = C(x_i)$$

- We look for an encoder *C** that achieves the goal of clustering data, according to some specific requirement of the algorithm and based on data pairs dissimilarities

Combinatorial Algorithms

- **Criterion**: assign to the same cluster similar/close data
- We may start from the following ”loss” or energy function (within class):

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})$$

$$C^* = \arg\min W(C)$$

Infeasible in practice!!!

K-Means Algorithm

It refers specifically to the Euclidean distance.

- initialise cluster centroids $m_k \quad k = 1, \ldots, K$ at random
- repeat until convergence
  - assign data to centroids $C(x_i) = \arg \min_{1 \leq k \leq K} \|x_i - m_k\|^2$
  - update centroids

K-means accounts at minimising the following functional

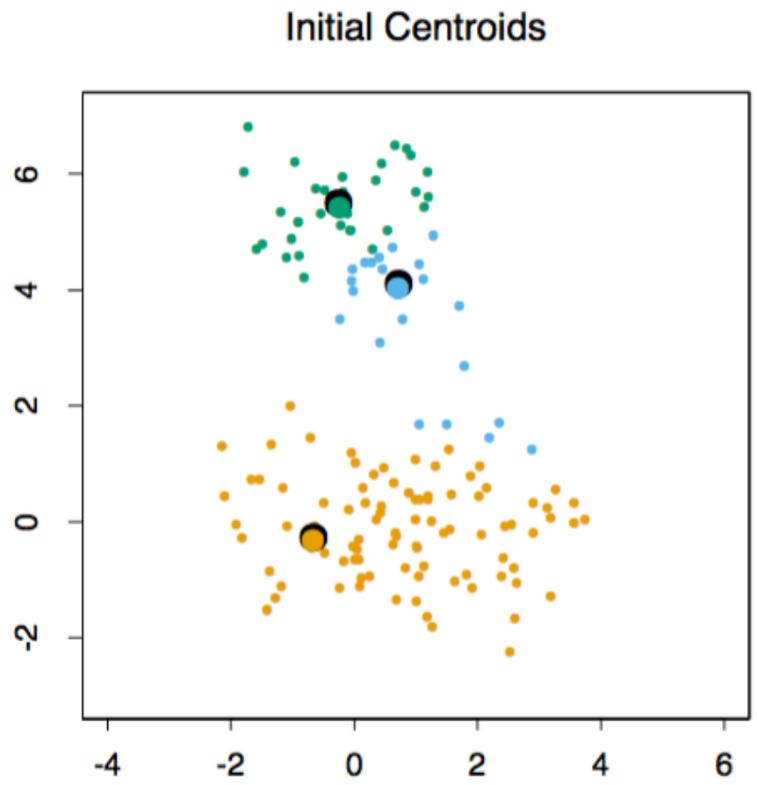
Convergence in practice

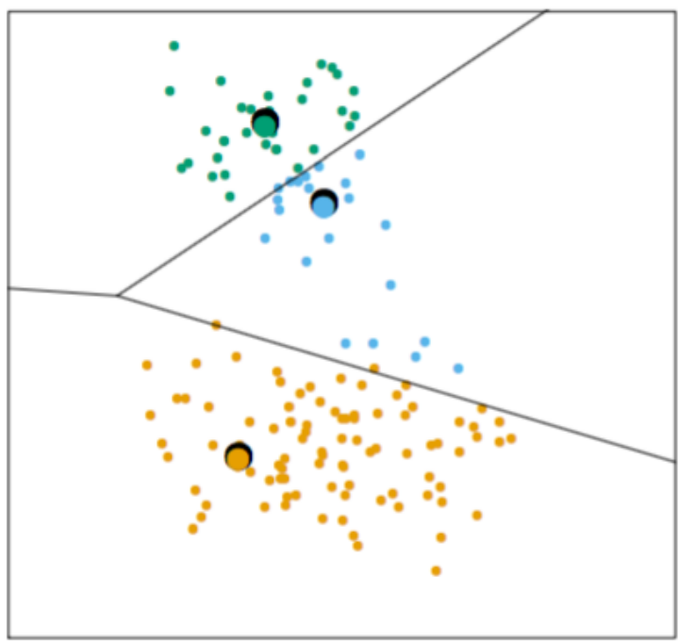
$$J(C, m) = \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - m_k\|^2$$

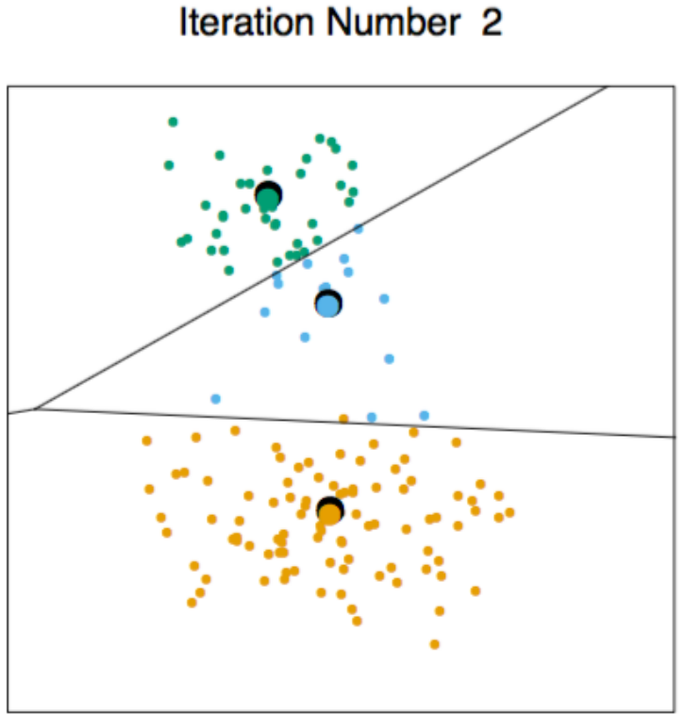
Non-convex

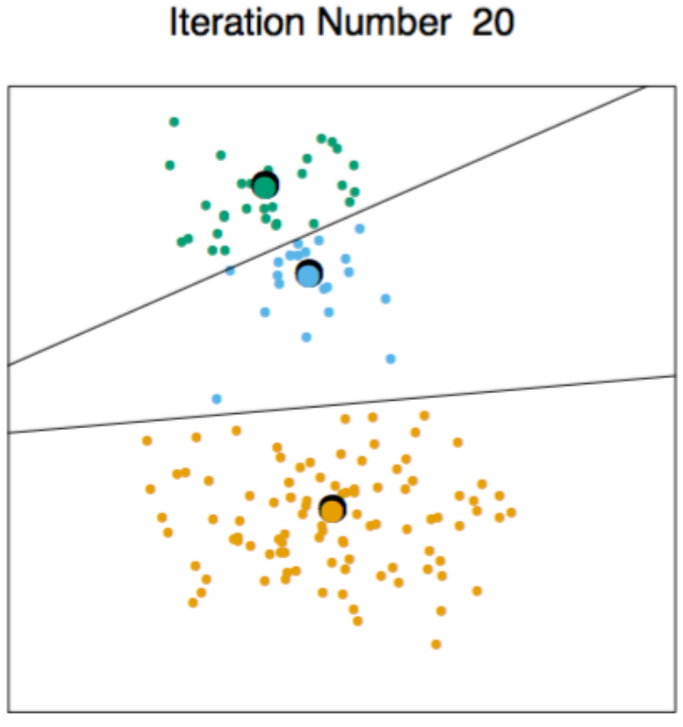
No guarantee for global minimum

K-Means Algorithm









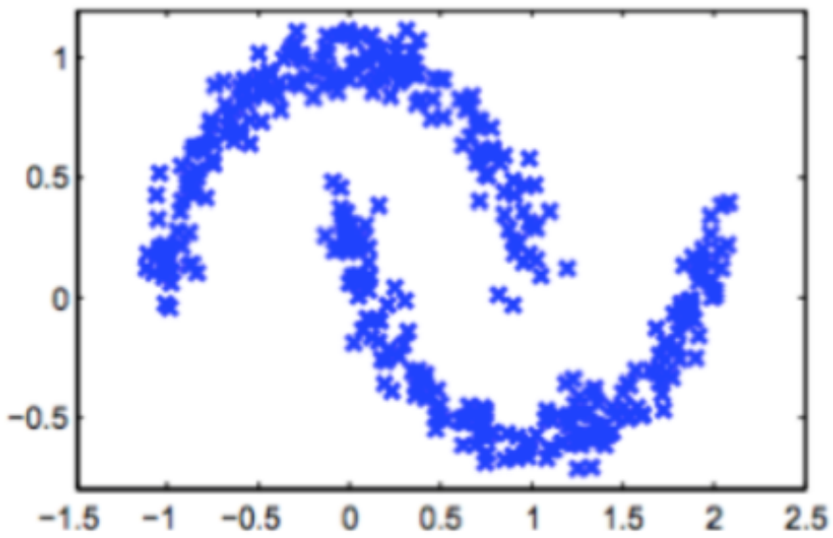
Example: Vector Quantisation

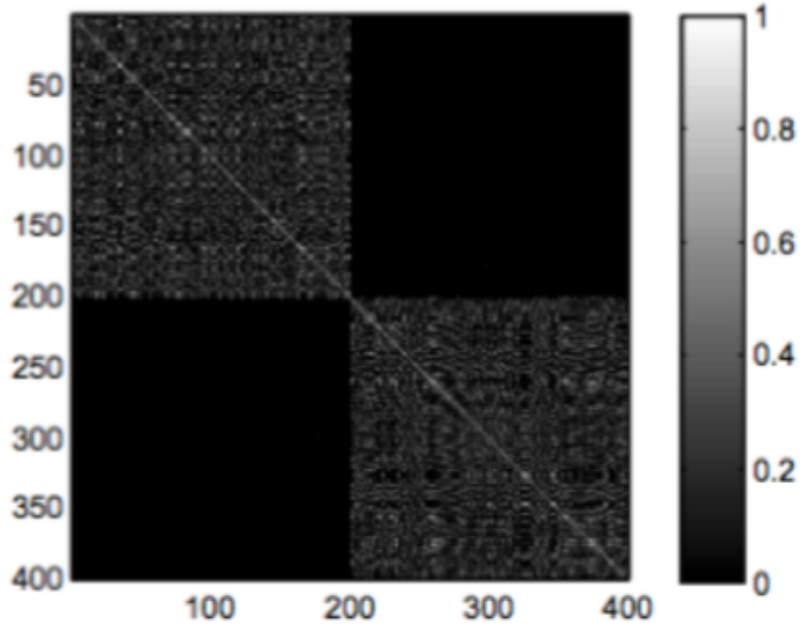
FIGURE 14.9. *Sir Ronald A. Fisher (*$1890 - 1962$*) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at* 8 *bits per pixel. The center image is the result of* 2×2 *block VQ, using* 200 *code vectors, with a compression rate of* 1.9 *bits/pixel. The right image uses only four code vectors, with a compression rate of* 0.50 *bits/pixel*

Spectral Clustering

A set of unlabelled data and some notion of similarity between data pairs

- We may represent them as a *similarity graph* G





- Clustering can be seen as a graph partitioning problem

End of Part IIIb

Clustering - Combinatorial Algorithms - Spectral Clustering

Acknowledgment

Timo Klock
Biomedical Computing
Department

Aslak Bergersen
Cardiac Modeling
Department

Sigurd Lekve
NTNU

Kristian Valen-Sendstad
Cardiac Modeling
Department