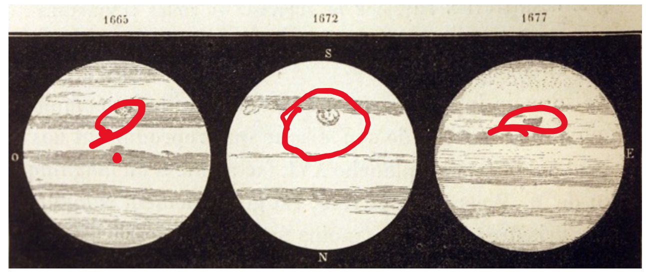
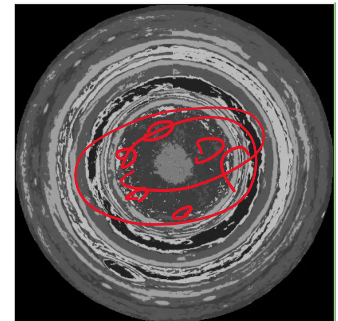
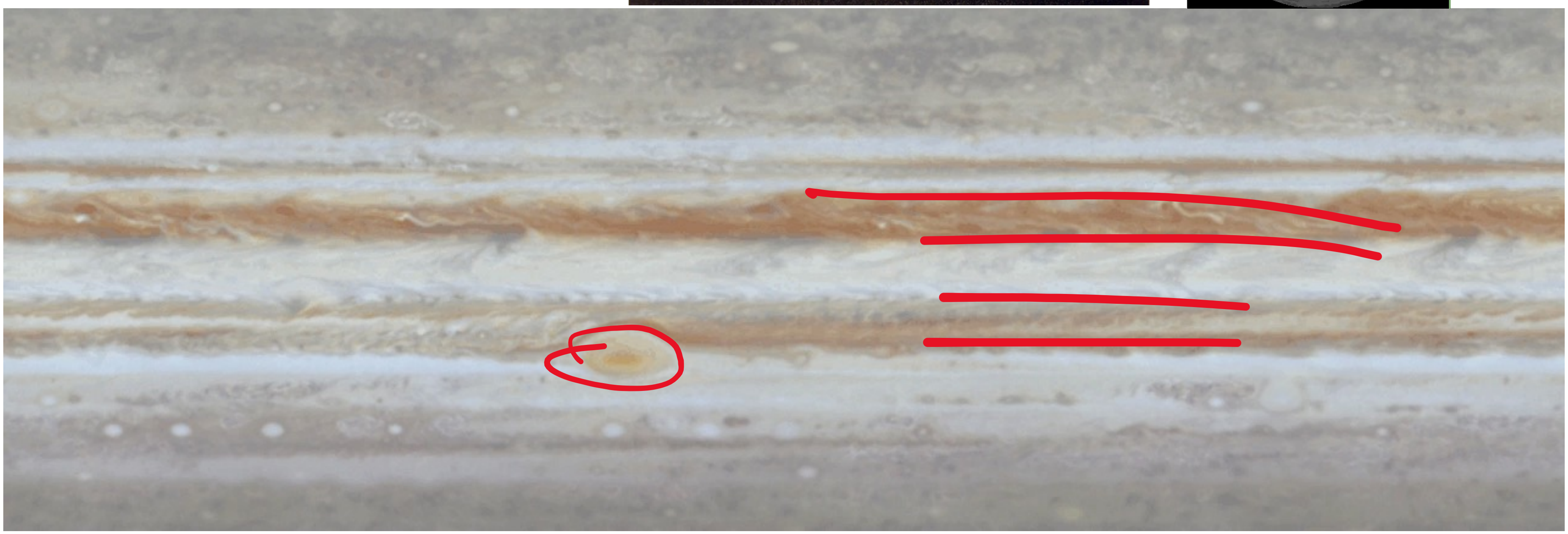
EE520 Data Driven Analysis of Complex Systems

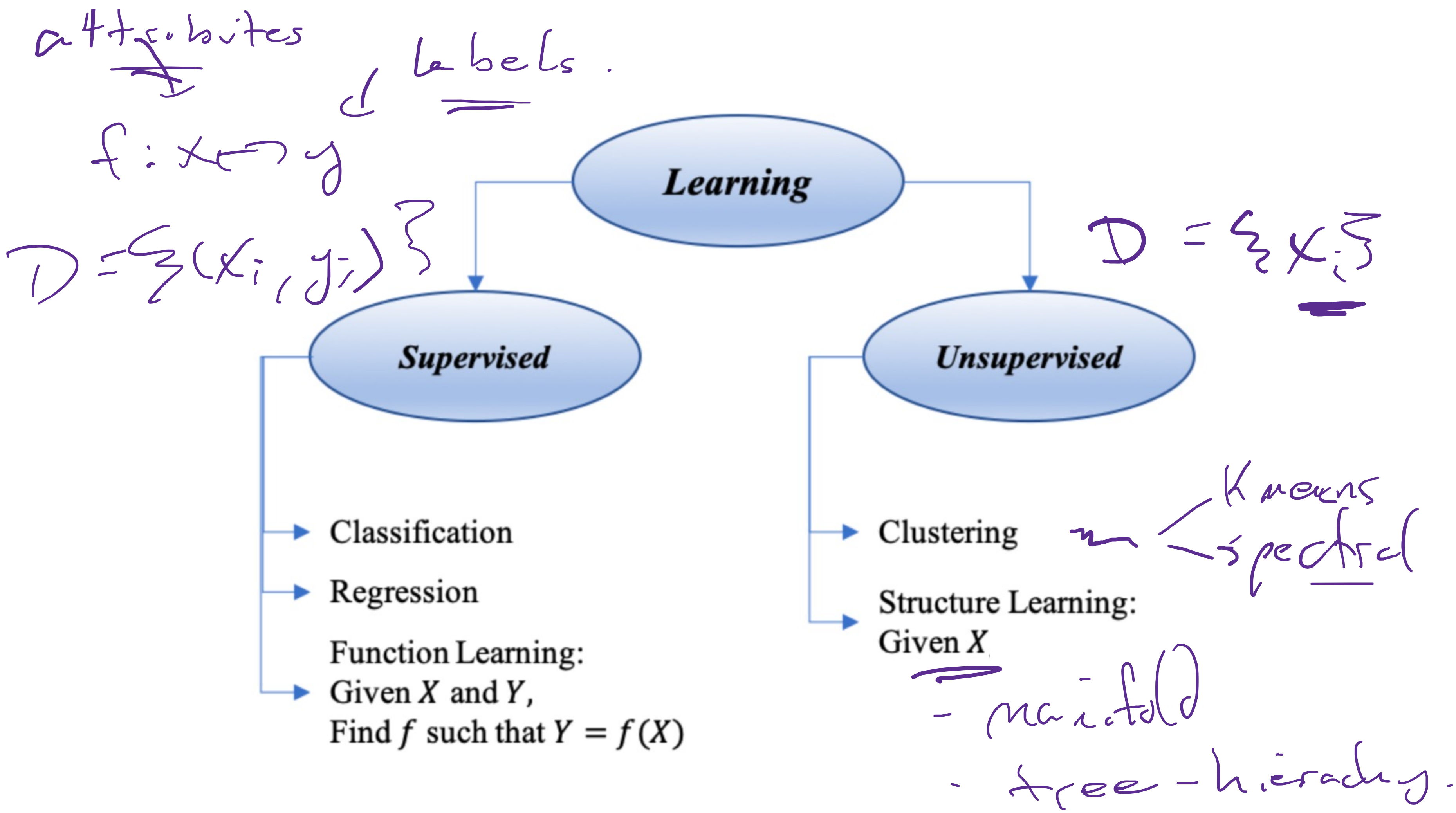
Erik Bollt





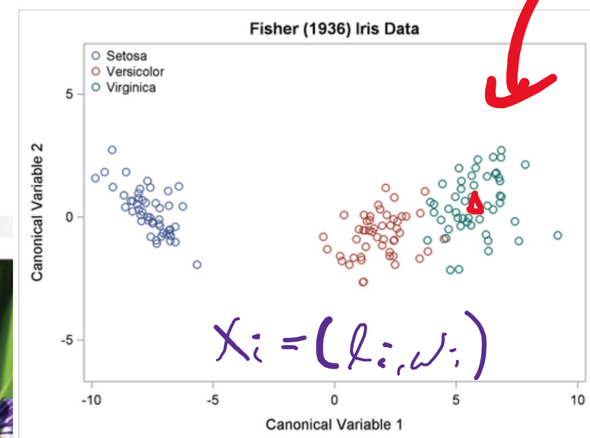


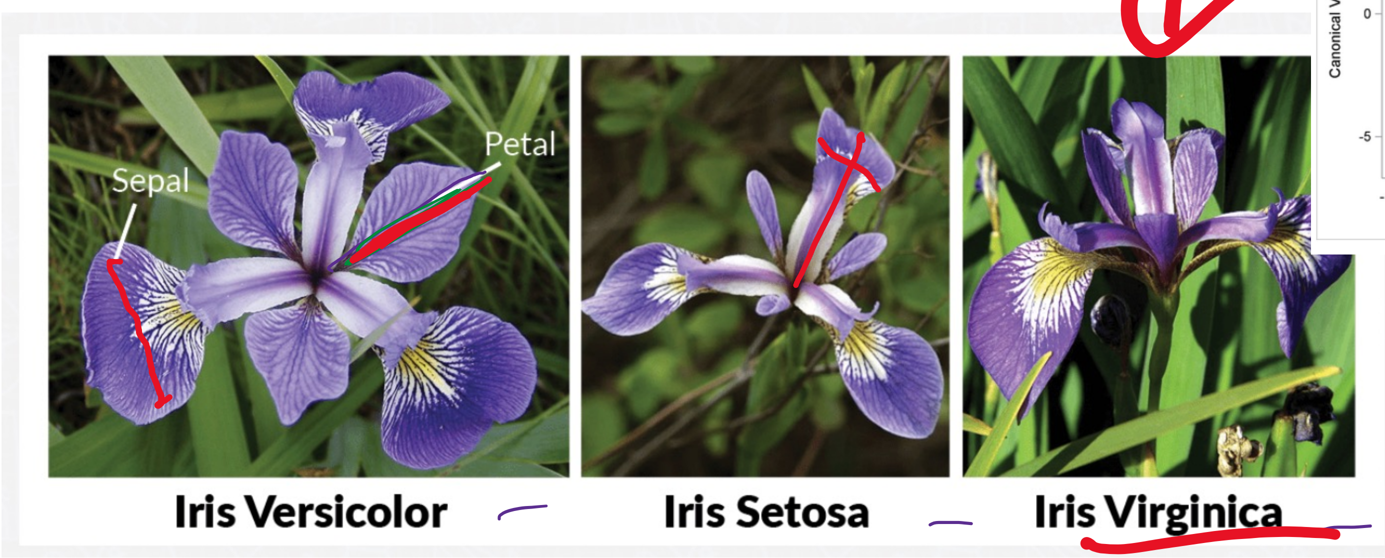
Ch 5 - Clustering and Classification

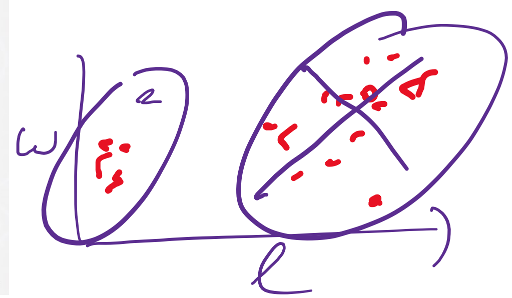

Learning
Supervised
Unsupervised
Classification
Regression
Function Learning:
Given X and Y,
Find f such that Y = f (X)
Clustering
Structure Learning:
Given X
attributes
labels
f : x → y
D = {(x_i, y_i)}
D = {x_i}
K means
spectral
manifold
tree-hierarchy

Discriminating Fisher's iris data by using the petal areas

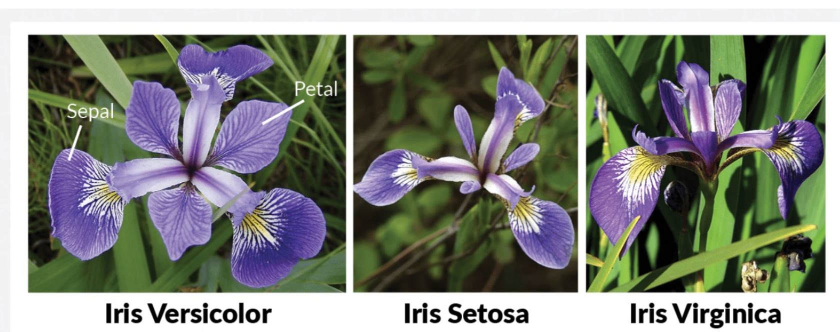
The Iris Dataset contains four features (length and width of sepals and petals) of 50 samples of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). These measures were used to create a linear discriminant model to classify the species. The dataset is often used in data mining, classification and clustering examples and to test algorithms.

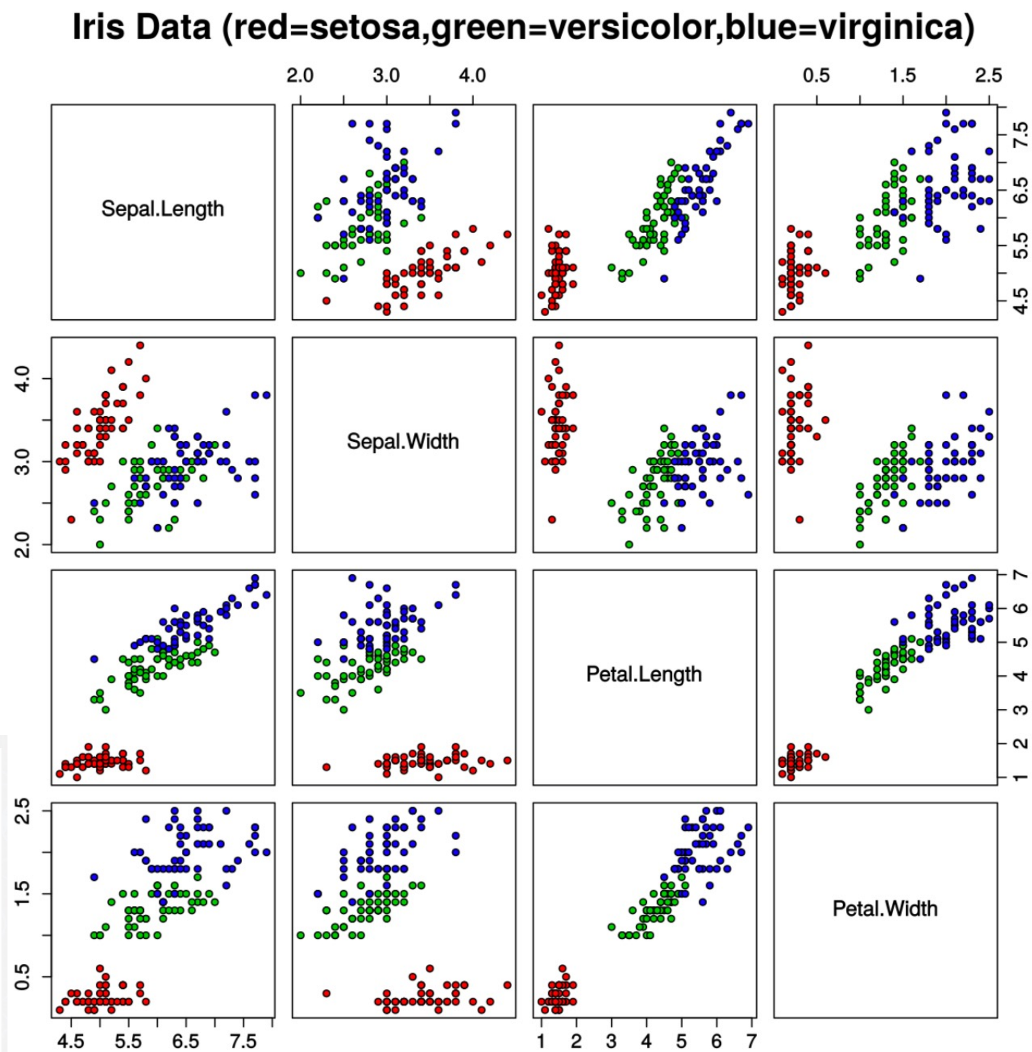






Ronald Fischer 1936





Data → Learning

$x_i = \{\text{income}, \text{zipcode}, \text{time of day}\}$

$y_i = \{0, 1, 2, 3\}$

$f: x \to y$

D_2 → supervised

D_1 → unsupervised

supervised

- learn f
  - regression $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$
  - classify if $y_i \in \mathbb{Z}_r = \{0, 1\}$, $r = 2$; $= \{\text{hot}, \text{cold}\}$

unsupervised

- no y_i
- no f
- ↳ just structure, just shape

$f: \mathbb{X} \to \mathbb{Y}$

$D_2 = \{(x_i, y_i)\}$; $x_i \sim \mathbb{X}$, $y_i \sim \mathbb{Y}$

vs. $z_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$

either

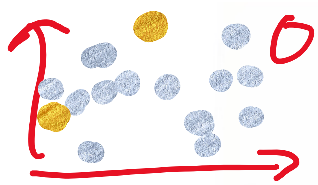
$D_1 = \{x_i\}_{i=1}^{n}$; $x_i \sim \mathbb{X}$

If we are going to do some machine learning – we had better get serious about what does learning mean?

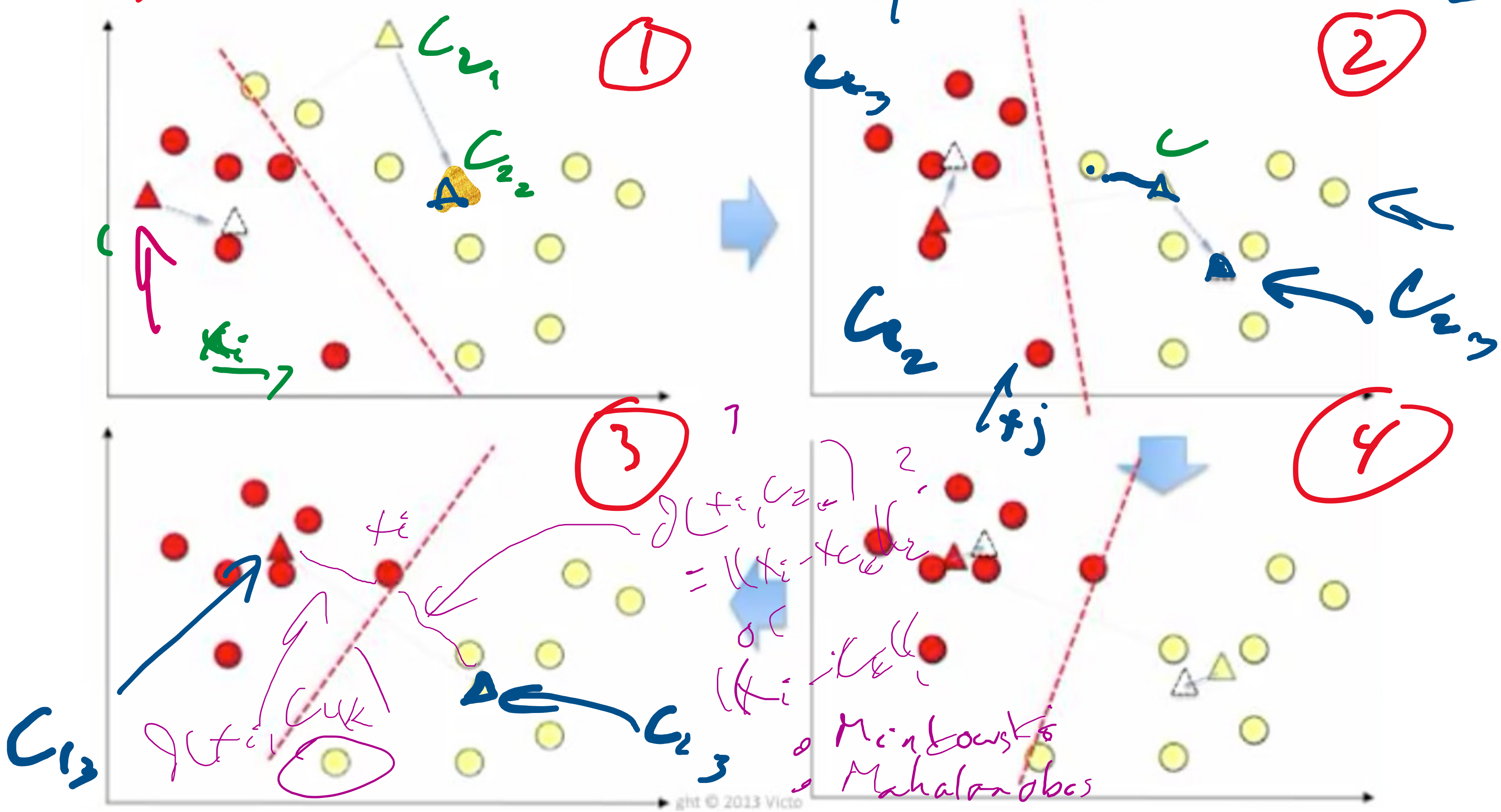
-Supervised discrete output (labels)   Classification
-Supervised continuous output   Regression

-Unsupervised discrete output (labels – cluster analysis)
-Unsupervised continuous output (ROM, manifold learning, density estimation)

Learning functions?
Learning structure?



K-means clustering example



https://files.realpython.com/media/centroids_iterations.247379590275.gif

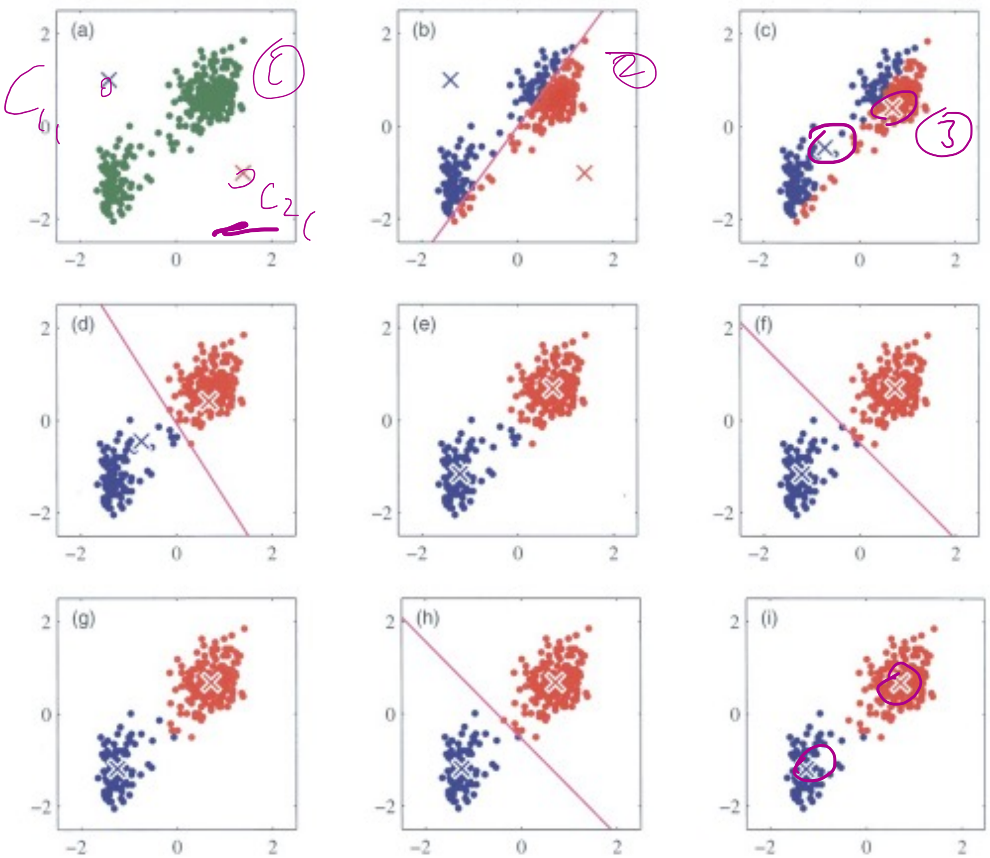


Illustration of K-means algorithm from [Bishop 2006]

Kmeans Convergence

Objective

$$\min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2$$

1. Fix μ, optimize C:

$$\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2 = \min_{c} \sum_{i}^{n} |x_i - \mu_{x_i}|^2$$

Step 1 of kmeans

2. Fix C, optimize μ:

$$\min_{\mu} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2$$

– Take partial derivative of μ_i and set to zero, we have

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

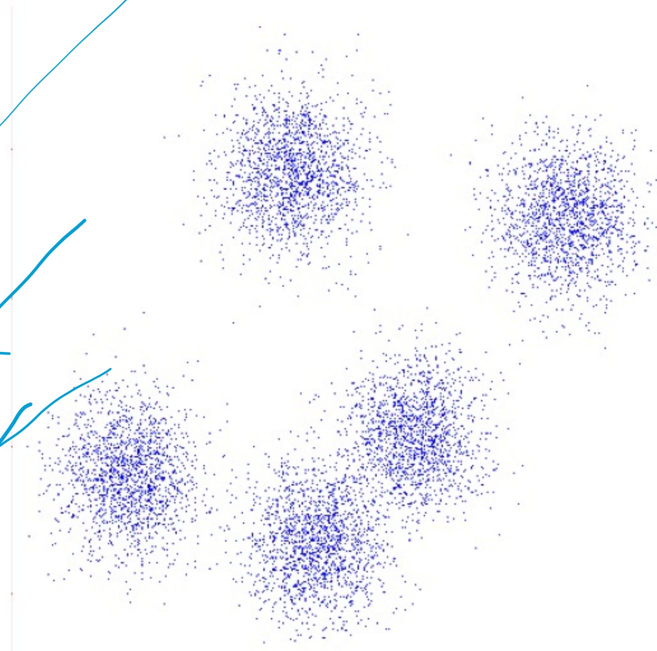
Step 2 of kmeans

Kmeans takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge

[Slide from Alan Fern]

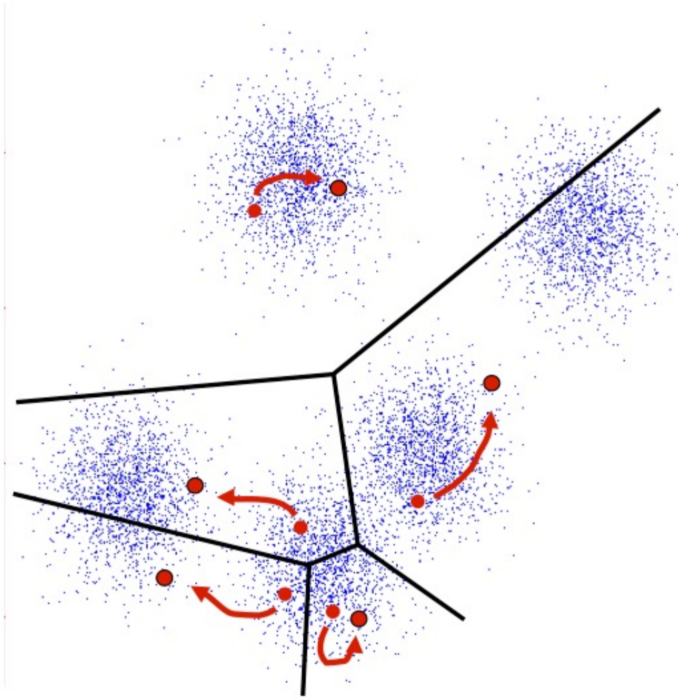
K-Means

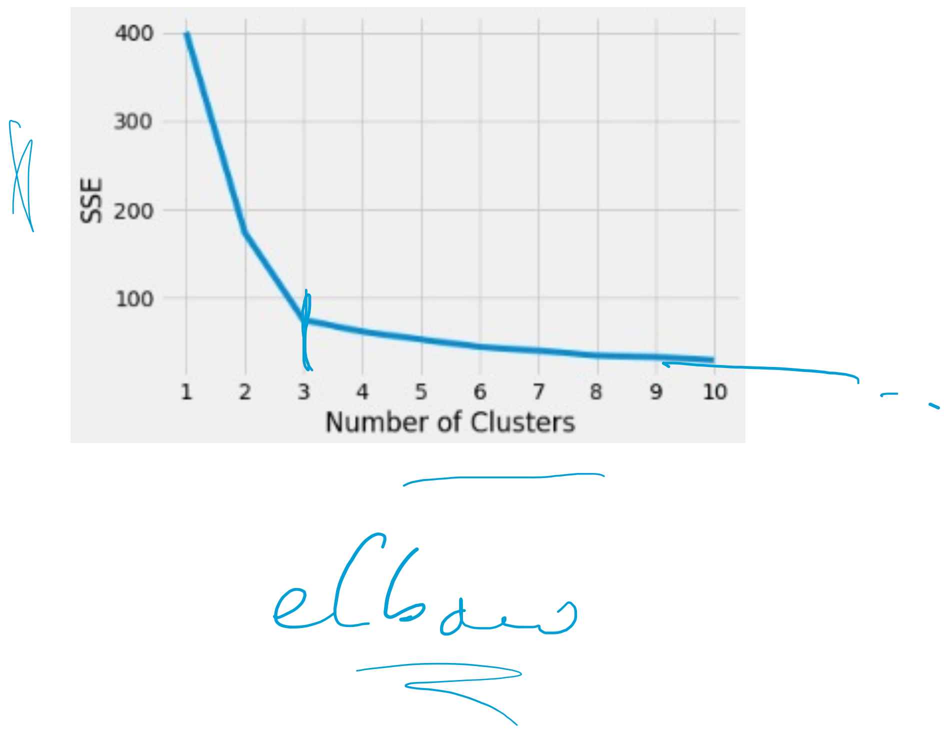
- An iterative clustering algorithm
  - Initialize: Pick *K* random points as cluster centers
  - Alternate:
    1. Assign data points to closest cluster center
    2. Change the cluster center to the average of its assigned points
  - Stop when no points' assignments change



K-Means

- An iterative clustering algorithm
  - Initialize: Pick *K* random points as cluster centers
  - Alternate:
    1. Assign data points to closest cluster center
    2. Change the cluster center to the average of its assigned points
  - Stop when no points' assignments change





400
300
200
100
SSE
1
2
3
4
5
6
7
8
9
10
Number of Clusters
elbow

$x = (x_1, x_2)$

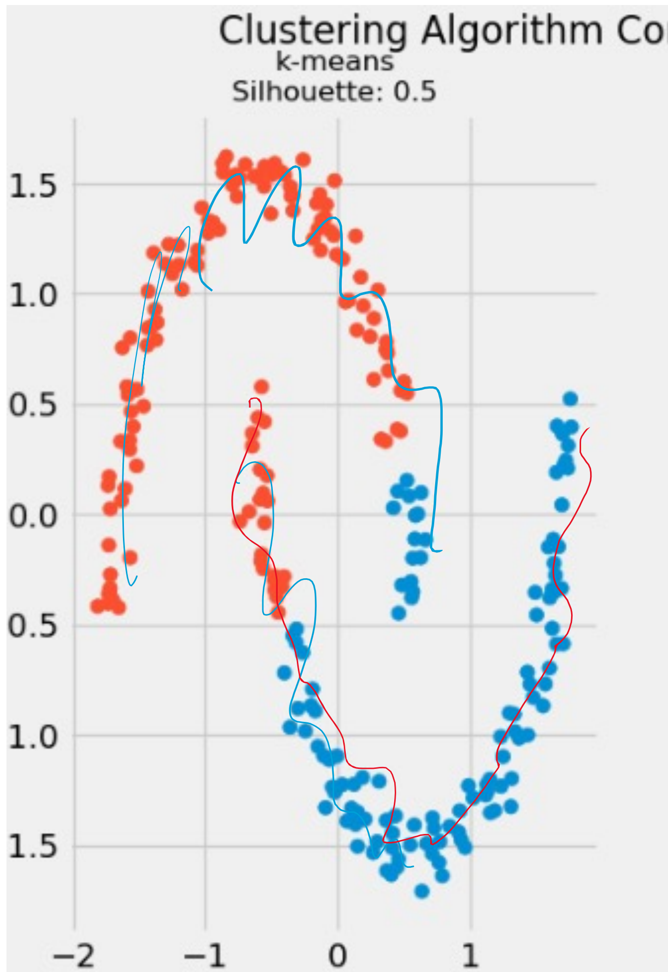
$C = (C_1, C_2)$

$Q(x, C)$

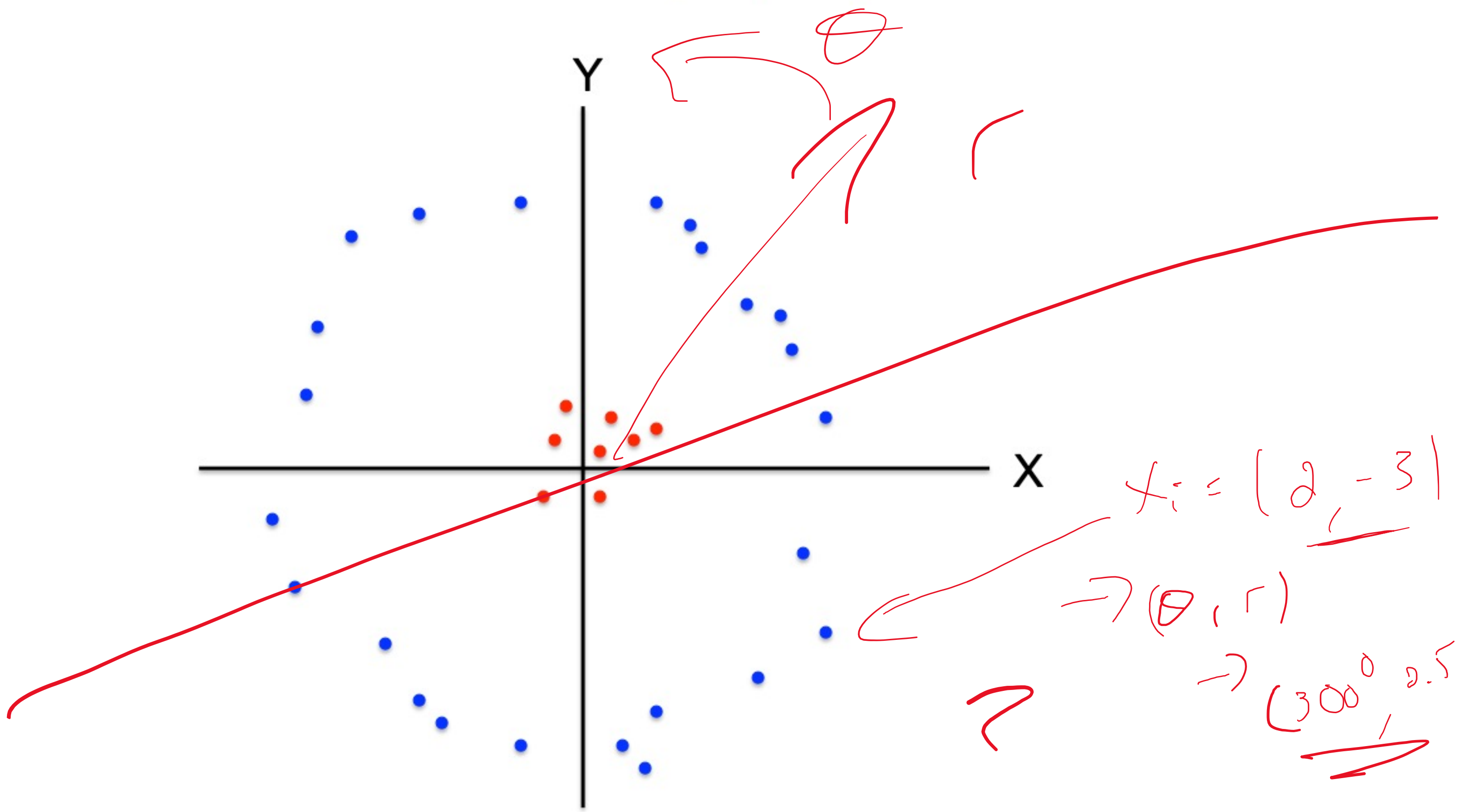
$= \sum_i (x_i - C_i)^2$

$= \sum (x_i - C_i)^4$

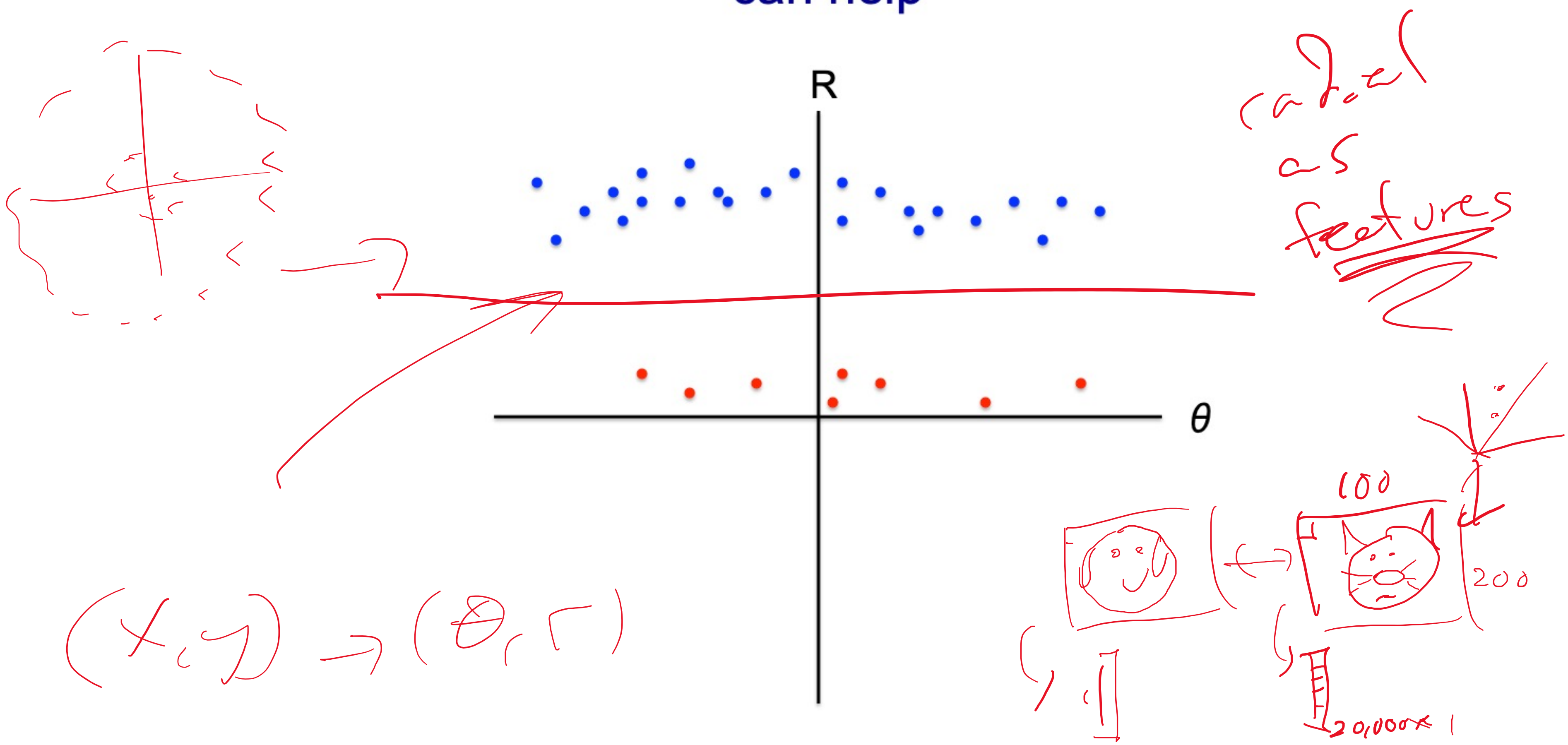
$\sum e^{t(x_i - C_i)}$?



K-means not able to properly cluster



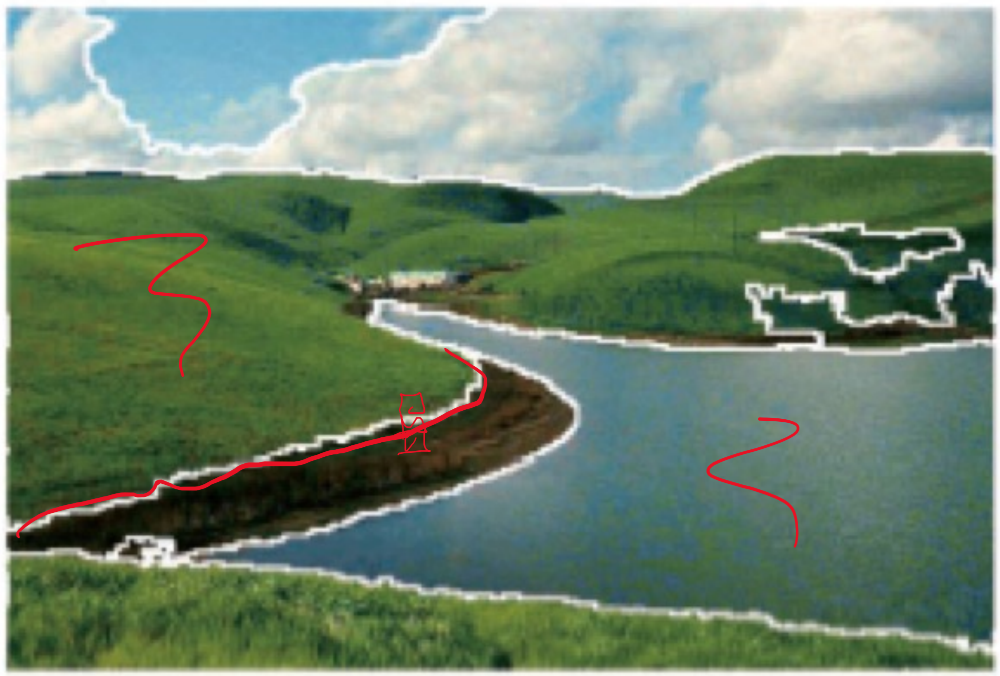
Changing the features (distance function) can help



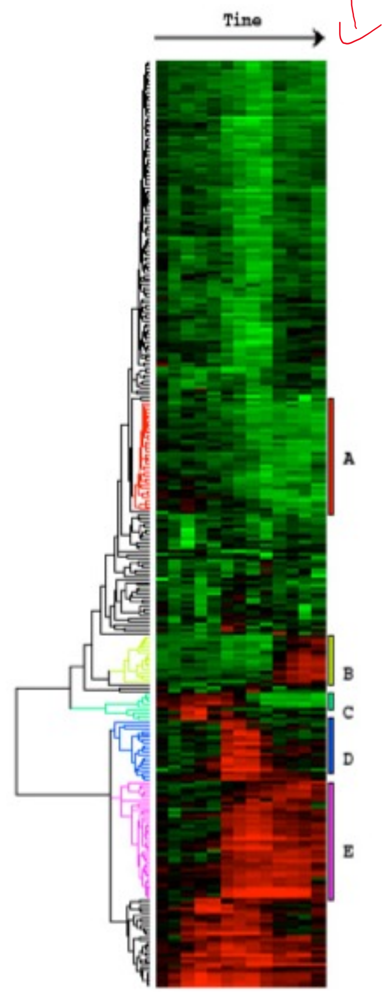
Example: K-Means for Segmentation

Image segmentation

Goal: Break up the image into meaningful or perceptually similar regions

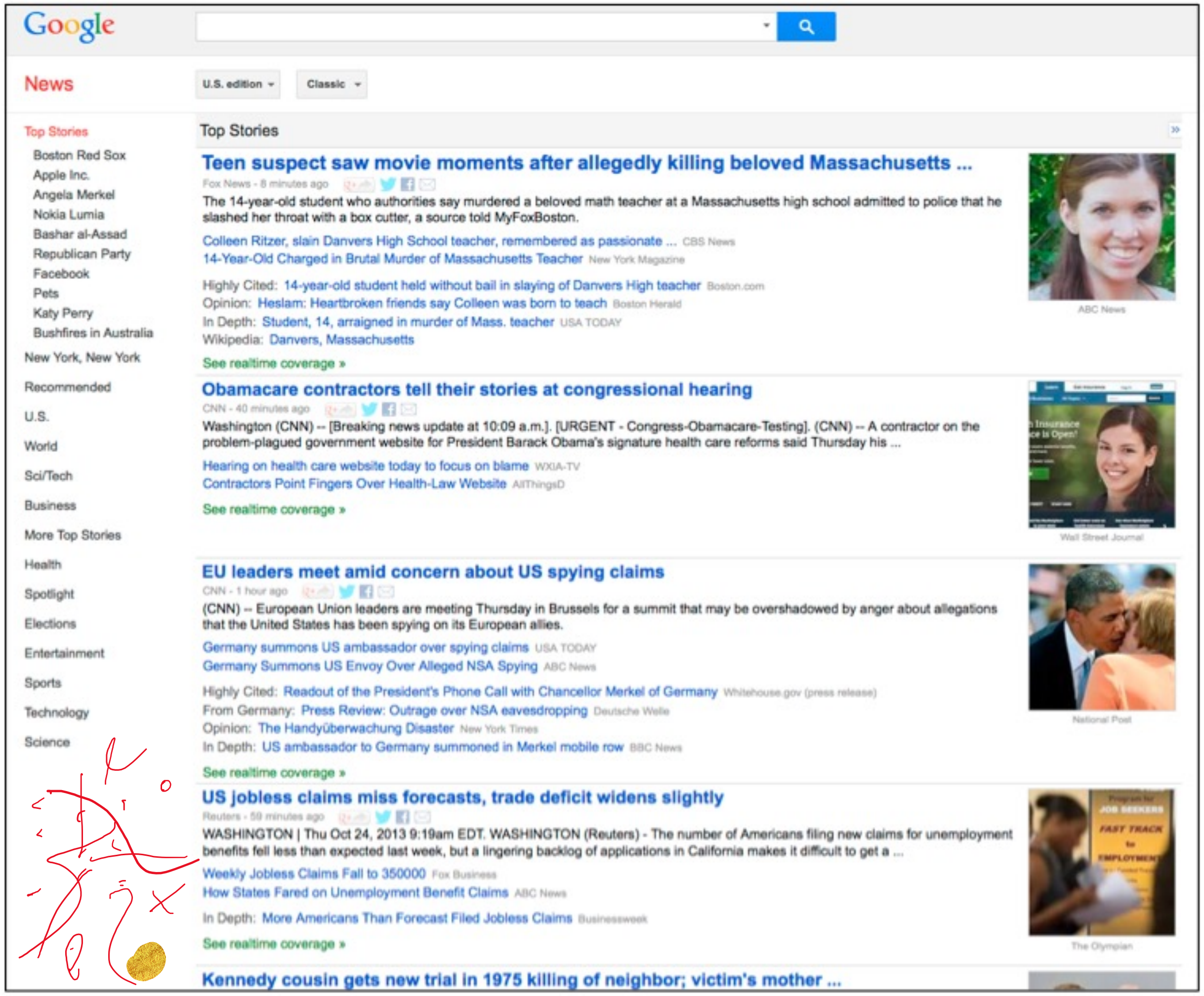


Clustering gene expression data



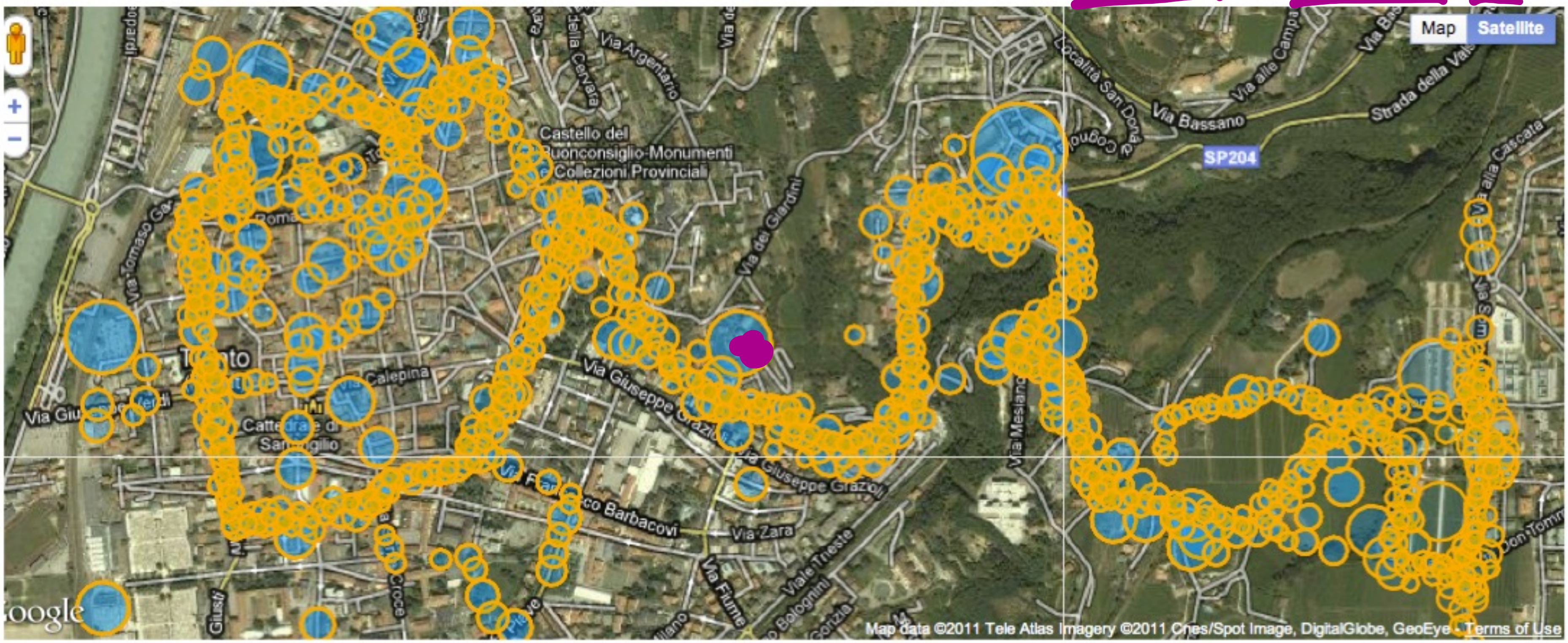
Eisen et al, PNAS 1998

Cluster news articles



Cluster people by space and time

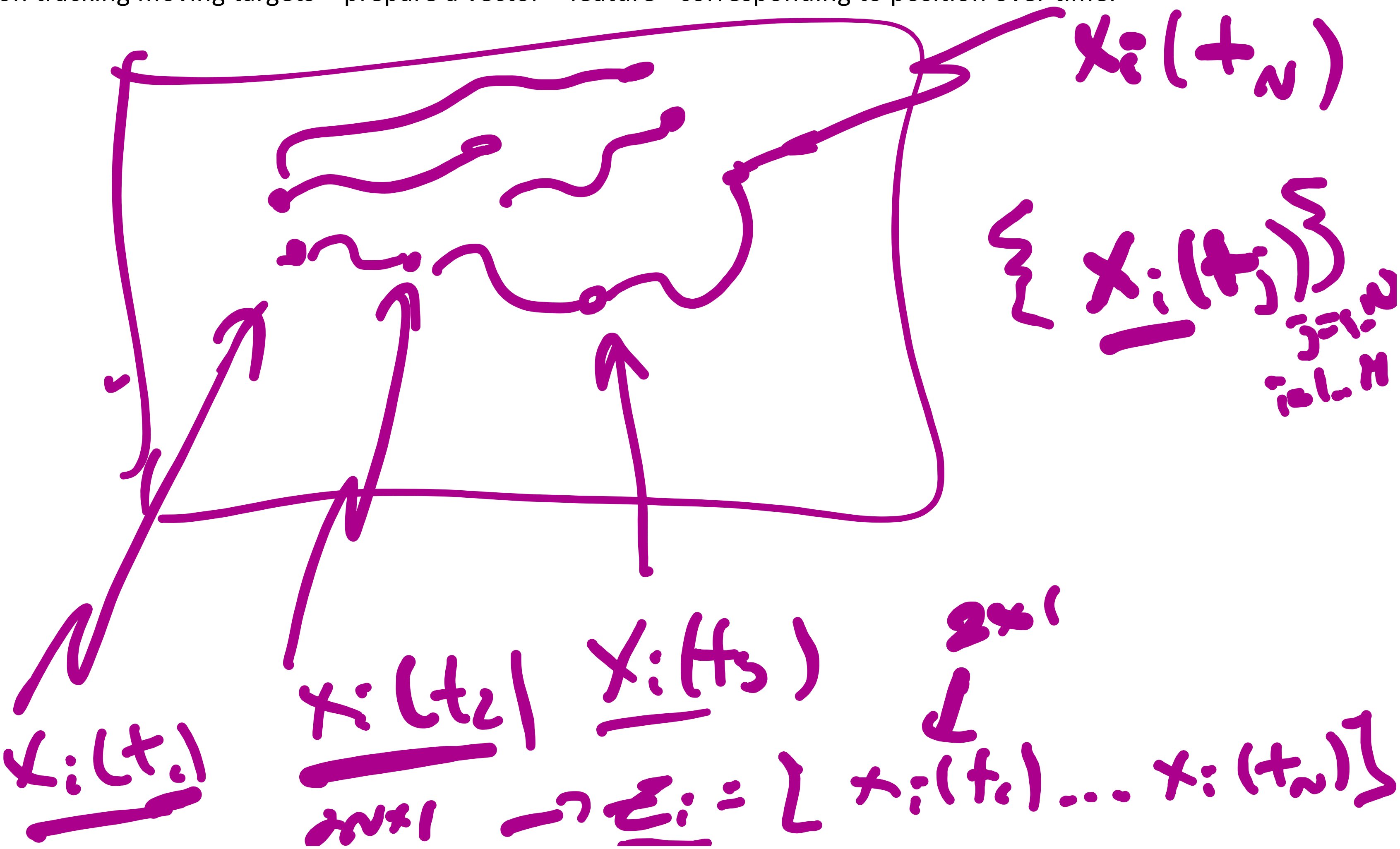
$z^i = (x_1^i, y_1^i, t_1^i), (x_2^i, y_2^i, t_2^i), (x_3^i, y_3^i, t_3^i)$



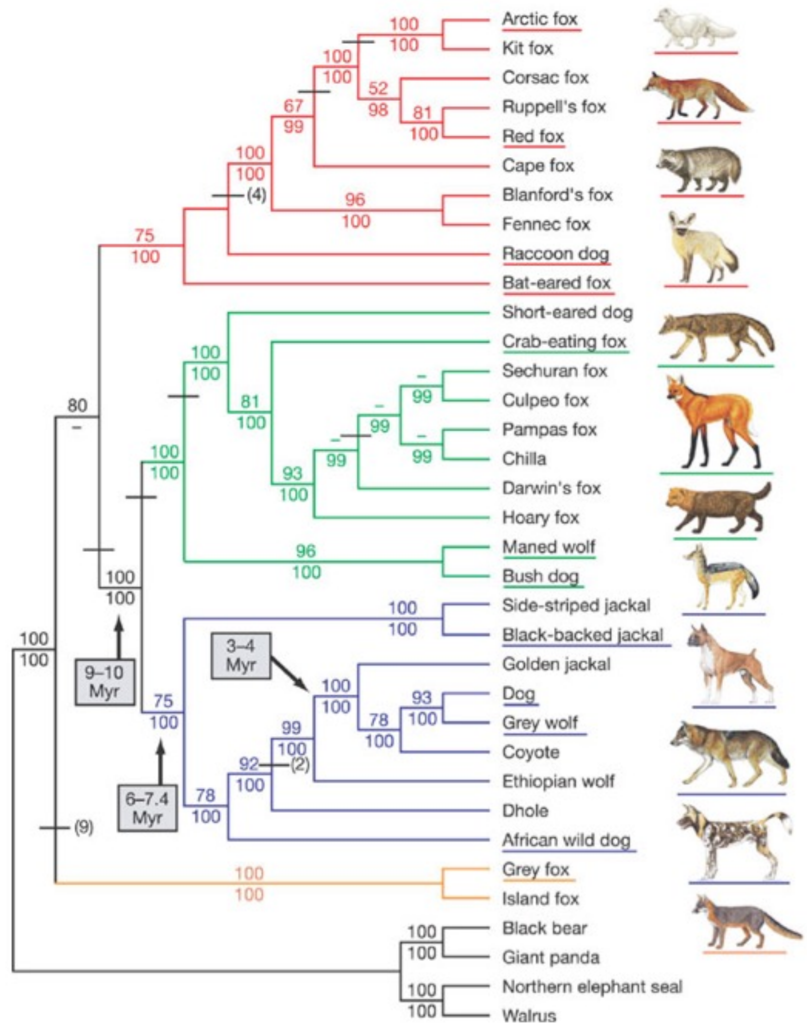
$\{z_i\}_{i=1}^N$

[Image from Pilho Kim]

Clustering on tracking moving targets – prepare a vector – feature - corresponding to position over time.



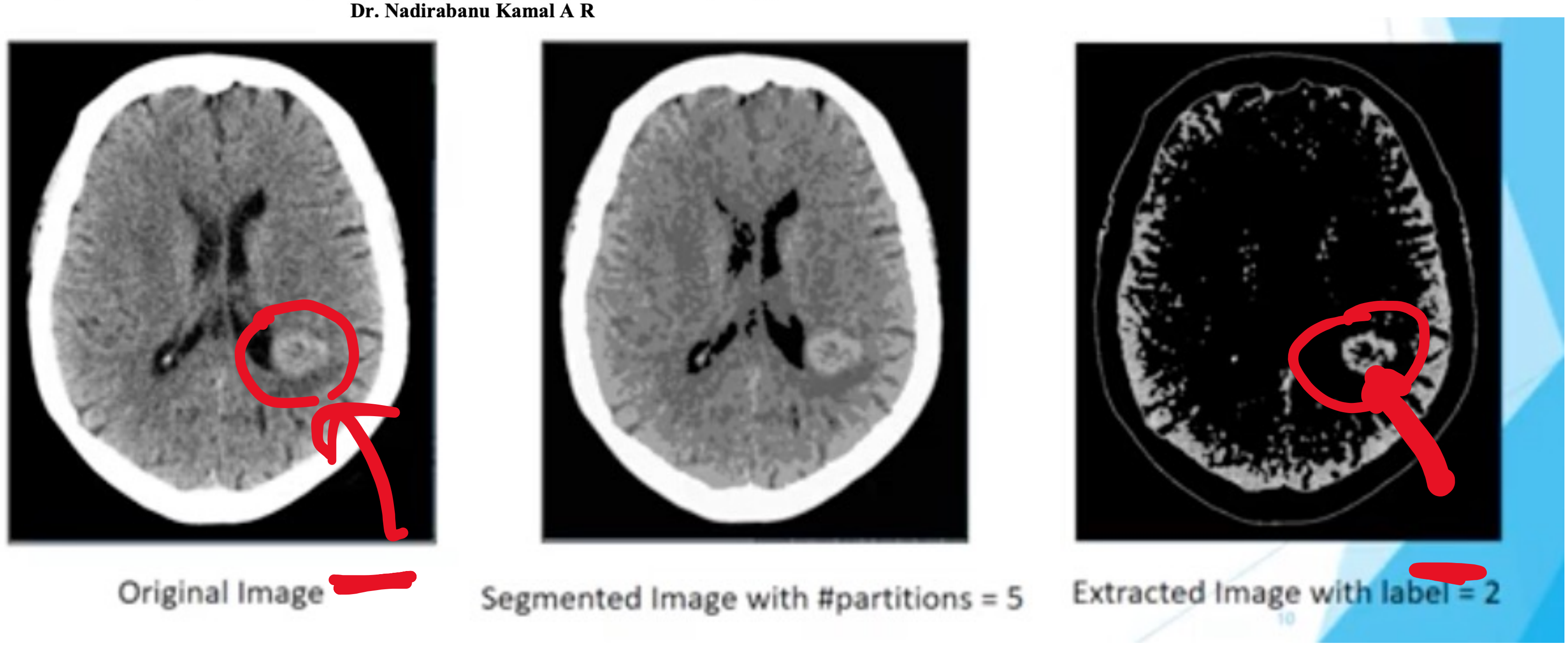
Clustering species ("phylogeny")



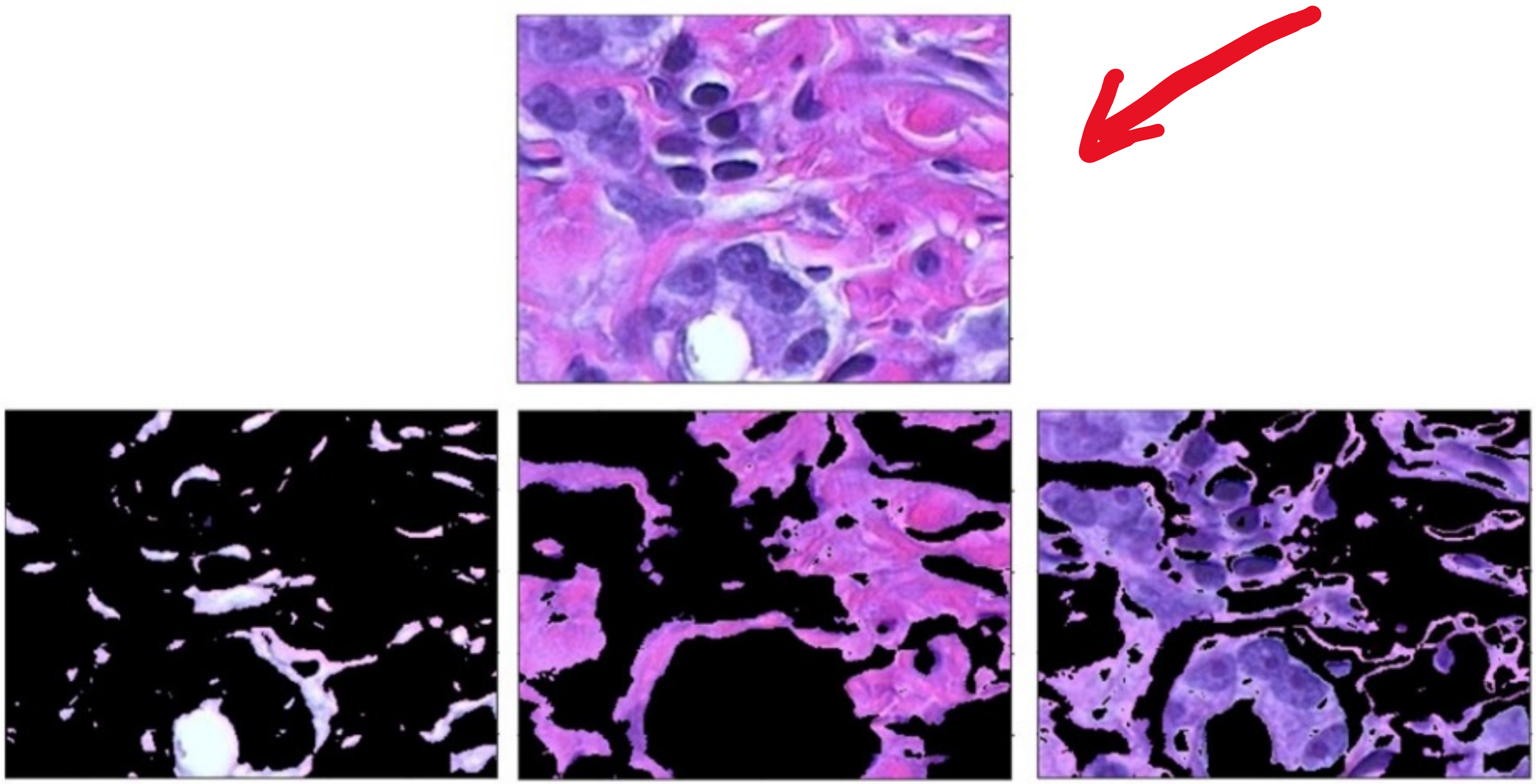
[Lindblad-Toh et al., Nature 2005]

Brain Tumor Detection and Identification Using K-Means Clustering Technique

Malathi R
Department of Computer Science, SAAS College, Ramanathapuram, Email: malapraba@gmail.com
Dr. Nadirabanu Kamal A R



Medical imaging – cancer tissue



IJEEE, Vol. 1, Issue 3 (June, 2014) e-ISSN: 1694-2310 | p-ISSN: 1694-2426

DETECTING AND COUNTING THE NO. OF WHITE BLOOD CELLS IN BLOOD SAMPLE IMAGES BY COLOR BASED K-MEANS CLUSTERING

[1]Neha Sharma, [2]Nishant Kinra
[1]Indira Gandhi Delhi Technical University for Women, New Delhi, India
[2]Deenbandhu Chhotu Ram University of Science and Technology, Haryana, India
[1]neha.sksharma@yahoo.co.in, [2]nishant.kinra@gmail.com

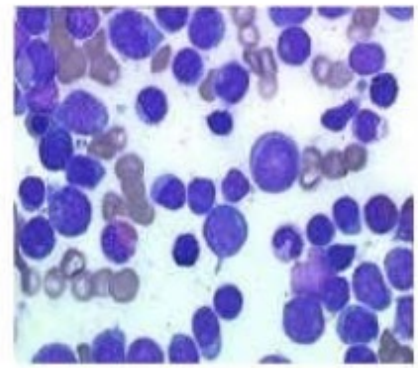


Fig (1): Example of Leukocytosis

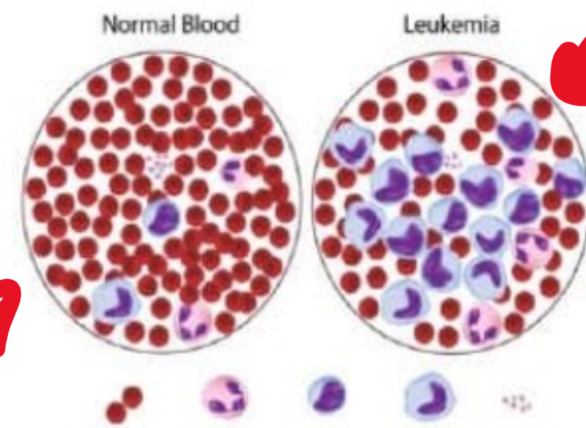


Fig (2): Comparison between Normal Blood and Leukemia

So that’s it for the time being with unsupervised learning, a few clustering methods – mostly kmeans,

But in your book are other interesting clustering methods in 5.4, 5.5, including tree methods and also Gaussian Mixture models that are very popular.

Later we will also develop a spectral clustering method that is very general and powerful.

Also in unsupervised learning later we will do “manifold learning”.

But for now…..

We transition to supervised learning

1st in 5.6 we will cover Fischer’s Linear Discriminant (LDA)

2nd in 5.7 on to support vector machines (SVM)

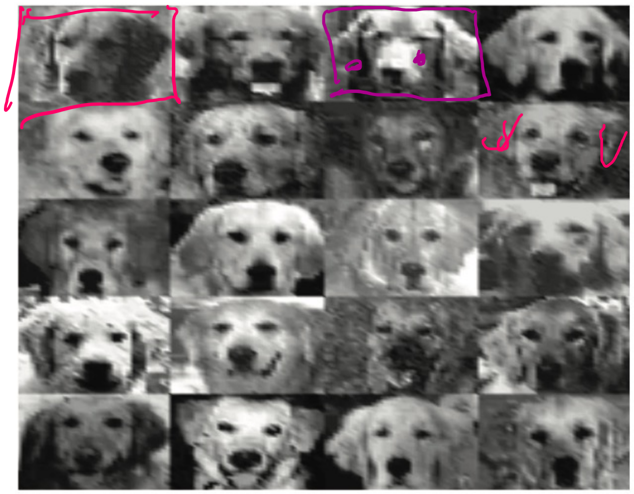
Then Chapter 6 a grandly popular method – artificial neural nets.

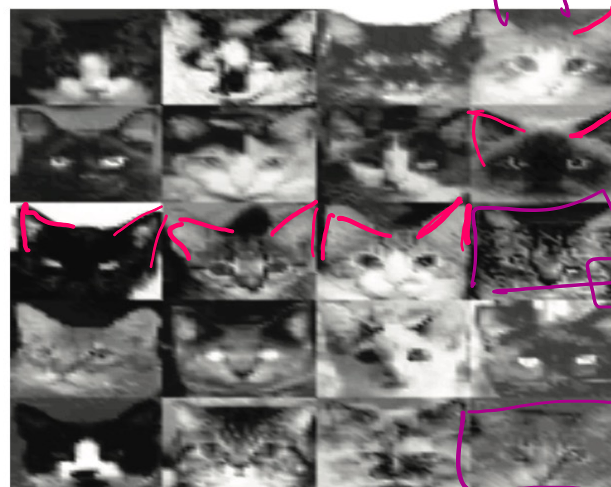
Supervised Learning – Cat or Dog?

y_i = Dog

x_i = ith picture

$[x_i]_{ppx}$





y_i = cat

160

100

200

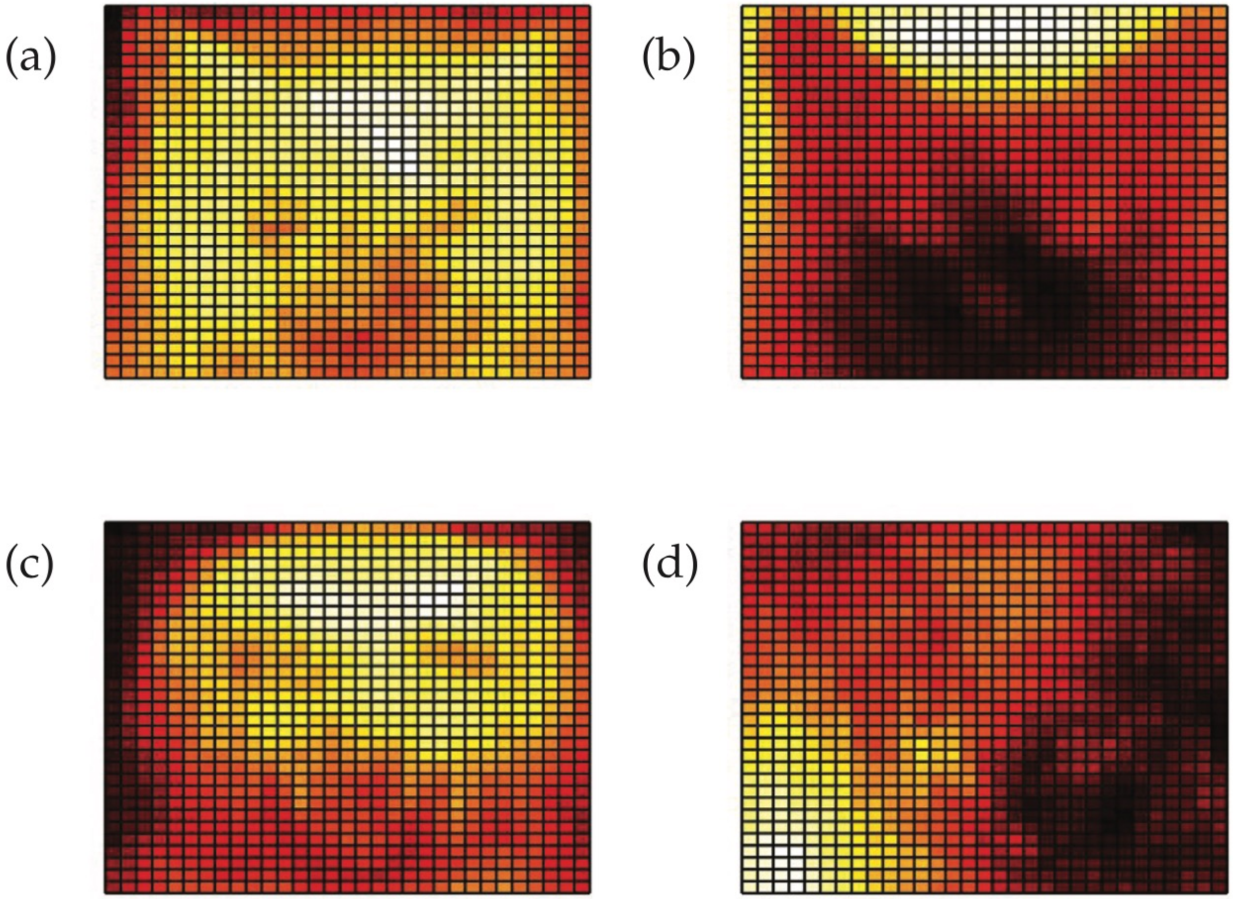
$D = \{(x_i, y_i)\}_{i=1}^{N}$

x_i = ith picture

y_i = ith label

$f: X \to Y \; ; \; f: \mathbb{R}^{20,000} \to \mathbb{Z}_2$

To distinguish cat's from dogs – first it will be more efficient if **we choose good (efficient) features**.



Wavelets then PCA (SVD) will work well.

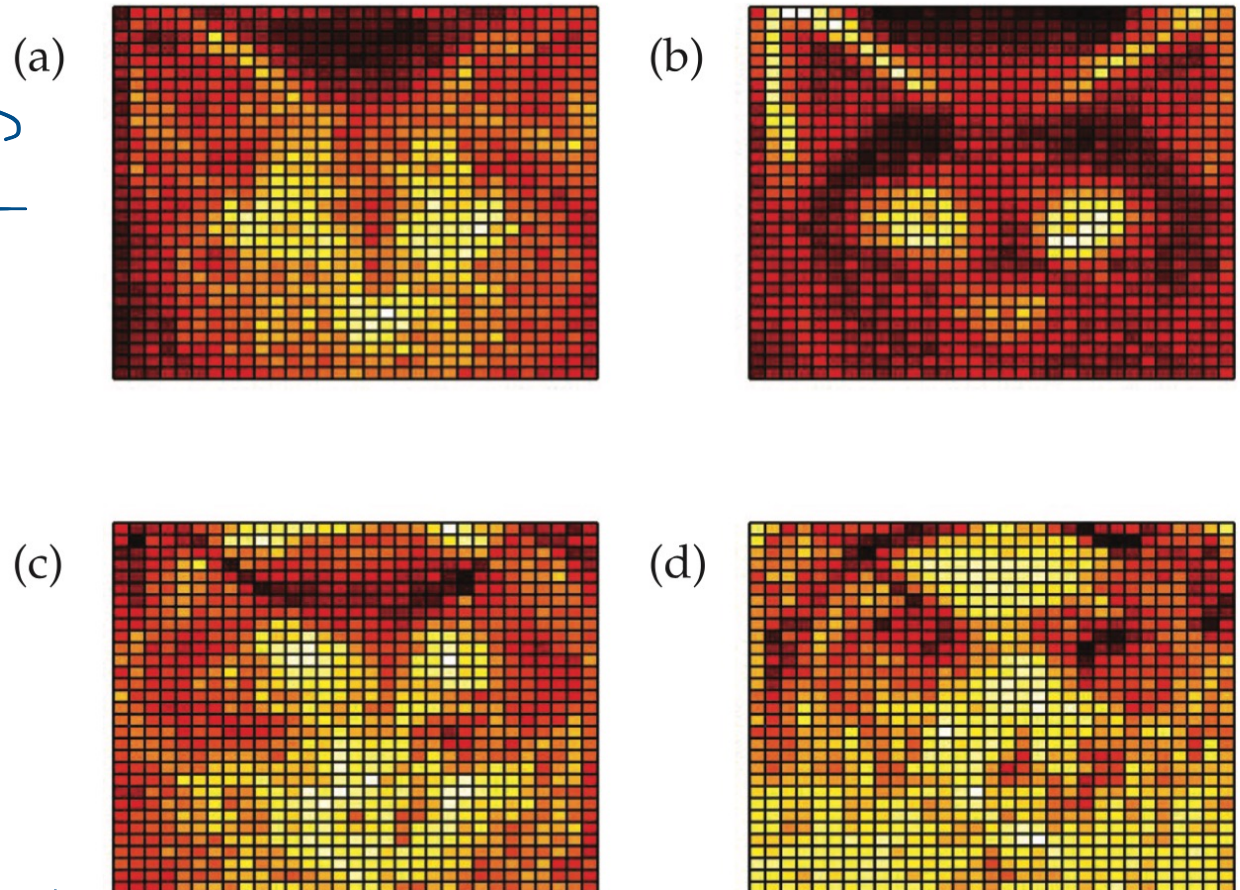
unsupervised.

$D = \{(x_i)\}$

structure

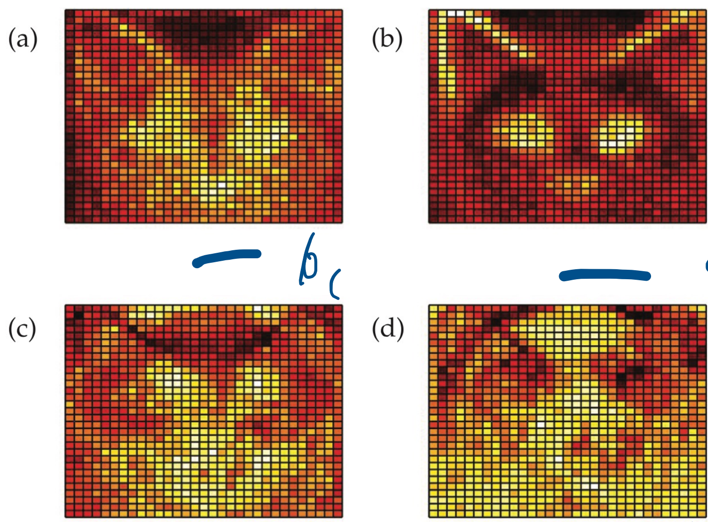
cluster

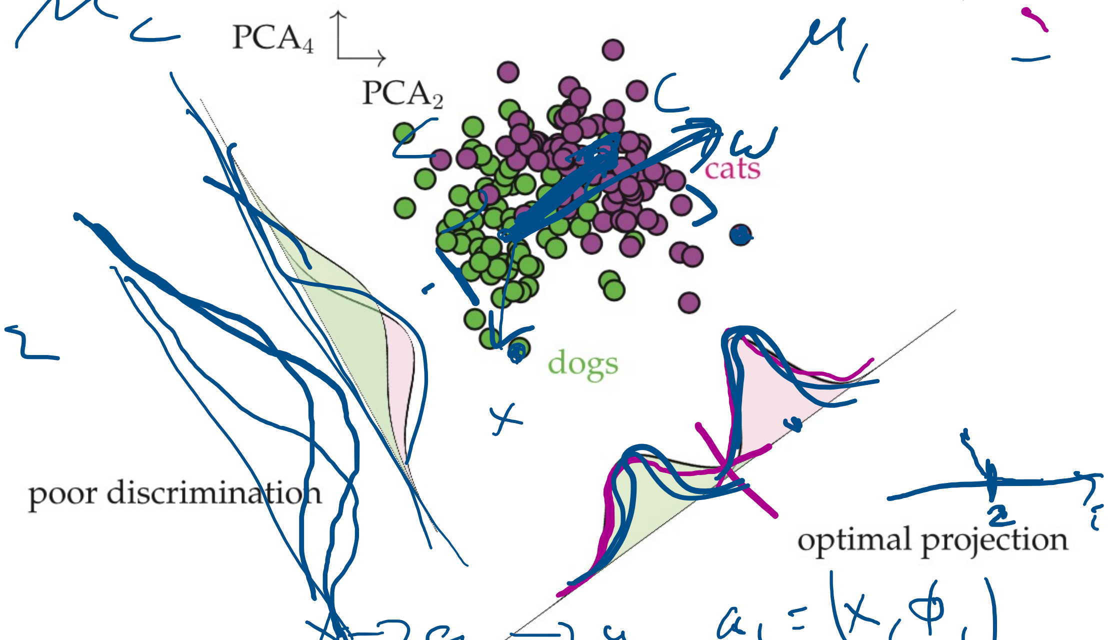
ROM



$D = \{(X_i, y_i)\}$ supervised – classify, $y = -1$ or 1.

Fischer’s Linear Discriminant Analysis LDA





$$\mathbf{w} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

where the scatter matrices for between-class $\mathbf{S}_B$ and within-class $\mathbf{S}_W$ data are given by

$$\mathbf{S}_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T$$

$$\mathbf{S}_W = \sum_{j=1}^{2} \sum_{\mathbf{x}} (\mathbf{x} - \mu_j)(\mathbf{x} - \mu_j)^T .$$

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$$

Rayleigh quotient whose solution can be found via the generalized eigenvalue problem

LDA – is a supervised learning problem of type – classification. Assumption – the data are Gaussian within class.

Suppose two classes of observations have means $\vec{\mu}_0, \vec{\mu}_1$ and covariances Σ_0, Σ_1. Then the linear combination of features $\vec{w}^{\mathrm{T}}\vec{x}$ will have means $\vec{w}^{\mathrm{T}}\vec{\mu}_i$ and variances $\vec{w}^{\mathrm{T}}\Sigma_i\vec{w}$ for $i = 0, 1$. Fisher defined the separation between these two distributions to be the ratio of the variance between the classes to the variance within the classes:

$$S = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{within}}} = \frac{(\vec{w}\cdot\vec{\mu}_1 - \vec{w}\cdot\vec{\mu}_0)^2}{\vec{w}^{\mathrm{T}}\Sigma_1\vec{w} + \vec{w}^{\mathrm{T}}\Sigma_0\vec{w}} = \frac{(\vec{w}\cdot(\vec{\mu}_1 - \vec{\mu}_0))^2}{\vec{w}^{\mathrm{T}}(\Sigma_0 + \Sigma_1)\vec{w}}$$

the scatter matrices for between-class $\mathbf{S}_B$ and within-class $\mathbf{S}_W$ data

$$\mathbf{S}_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T$$

$$\mathbf{S}_W = \sum_{j=1}^{2}\sum_{\mathbf{x}}(\mathbf{x} - \mu_j)(\mathbf{x} - \mu_j)^T \,.$$

$$\mathbf{w} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}$$

$$\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}$$

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \quad ; \quad w^* = \arg\max_w J(w)$$

I

$$\min \; -\frac{1}{2} w^T S_B w$$

$$\text{s.t. } w^T S_W w = 1$$

$$I \Rightarrow II$$

$$\mathcal{L} = -\frac{1}{2} w^T S_B w + \frac{\lambda}{2}\left(w^T S_W w - 1\right)$$

$$(\mu_2 - \mu_1) \cdot v$$

$$= (\mu_2 - \mu_1)^T v$$

$$= v \cdot (\mu_2 - \mu_1)$$

$$= v^T (\mu_2 - \mu_1)$$

$$\left((\mu_2 - \mu_1) \cdot v\right)^2 = v^T \left[(\mu_2 - \mu_1)^T \cdot (\mu_2 - \mu_1)\right] v$$

let

$$\mathcal{L} = \frac{1}{2} w^T S_B w + \frac{1}{2}\lambda (w^T S_W w - 1)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = 0; \quad \frac{\partial \mathcal{L}}{\partial w_2} = 0, \dots; \quad \nabla_w \mathcal{L} = 0$$

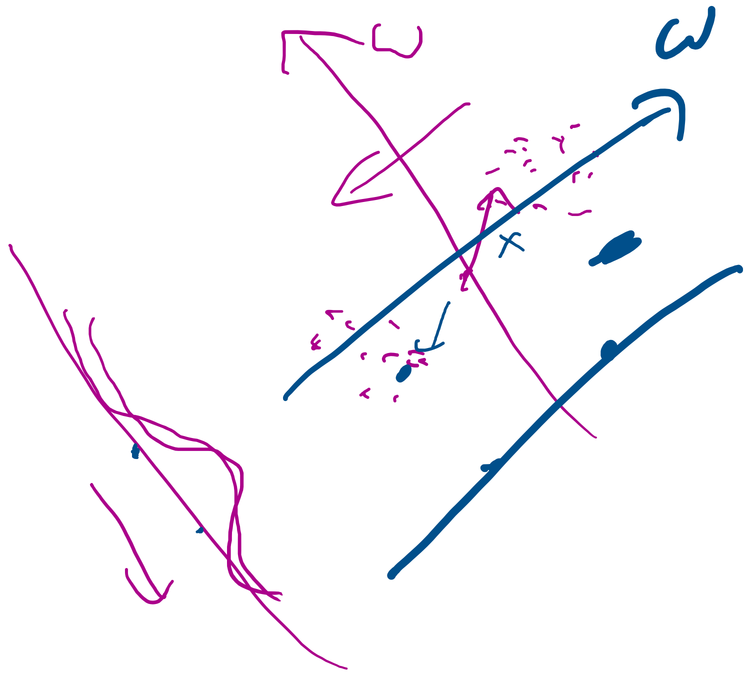
KKT

$$\nabla_w \mathcal{L} = -S_B w + \lambda S_W w = 0$$

$$S_B w = \lambda S_W w$$

$$\nabla \left(\frac{1}{2} w^T w\right) = w$$

$Ax = \lambda Bx$ generalized eigenvector/value statement!



$f : \mathcal{X} \to \mathcal{Y}$

$f : \mathcal{X} \to \ ?$

(x, ω)

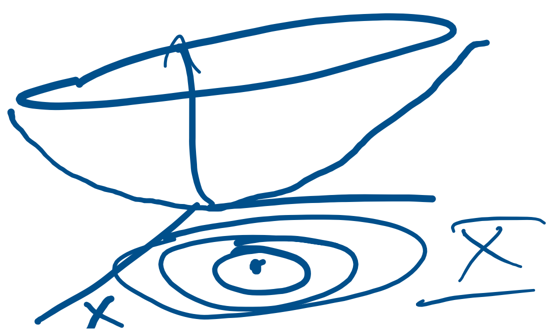
KKT:

$$x^* = \arg\min_x f(x)$$

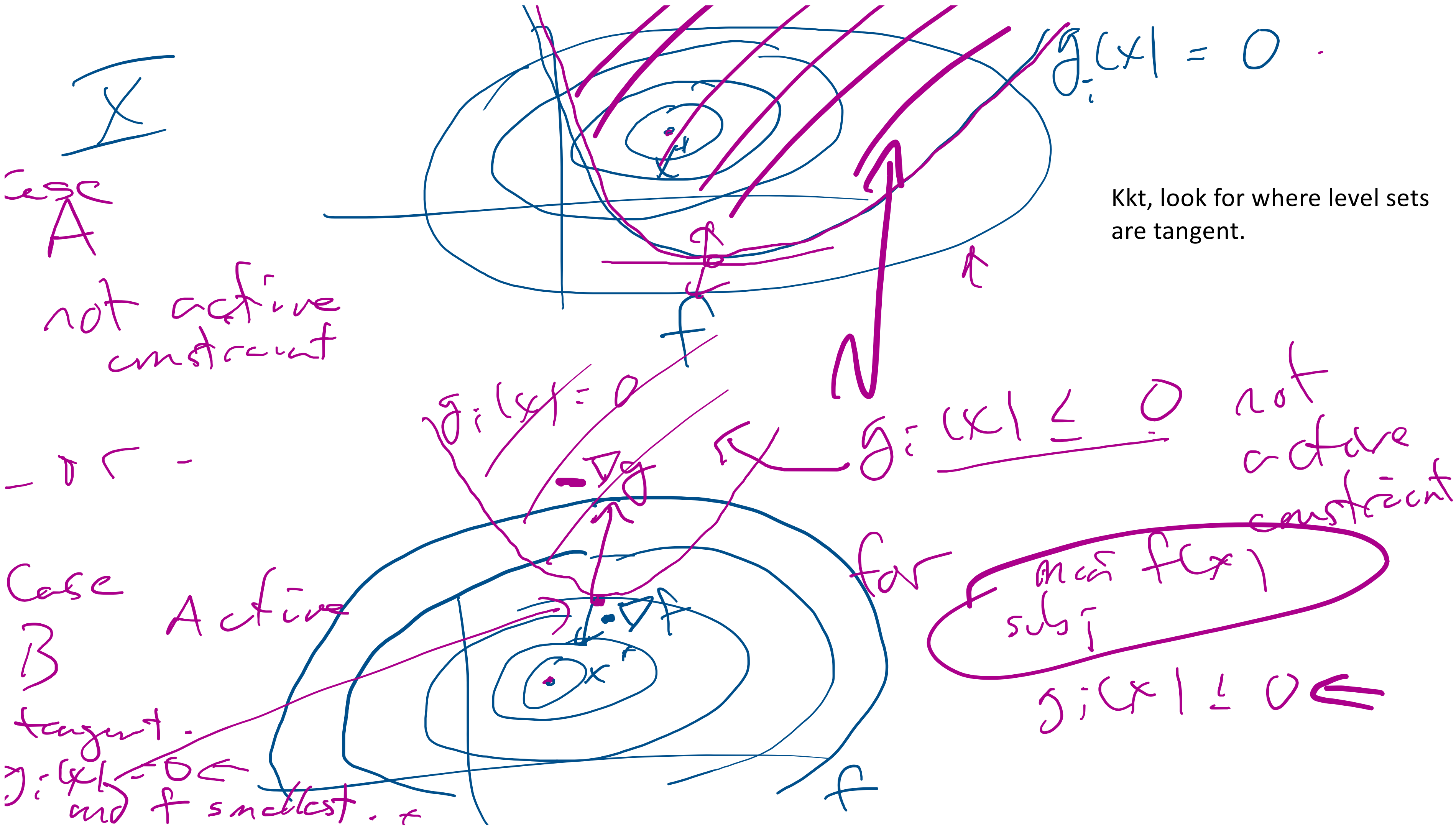
$$\text{subj. } h_i(x) = 0, \ \forall i = 1, \dots, m; \quad \vec{h}(\vec{x}) = \vec{0}$$

$$g_i(x) \leq 0; \ \forall i = 1, \dots, n \qquad \sout{\vec{g}(\vec{x}) \leq \vec{0}}$$

$$x^* = \arg\min_{\substack{x \\ \lambda, \mu}} \mathcal{L}(x, \lambda, \mu) = \arg\min_x f(x) + \sum_{i}^{m} \lambda_i h_i(x) + \sum_{j}^{n} \mu_j g_j(x)$$



$$f : X \to \mathbb{R}$$



Kkt, look for where level sets are tangent.

KKT $\nabla g \parallel \nabla f$ at constrained opt.

$$\lambda \nabla g = -\nabla f$$

$$\nabla f + \lambda \nabla g = 0 \quad \ldots\ldots \text{Lagrange multiplier.}$$

Equality constr.

$$\rightarrow \nabla_x f + \sum \nabla_x \lambda_i h_i(x) + \sum \mu_i \nabla_x g_i(x) = 0$$

- stationarity. ←
- ineq. constr true.

Haar Wavelet

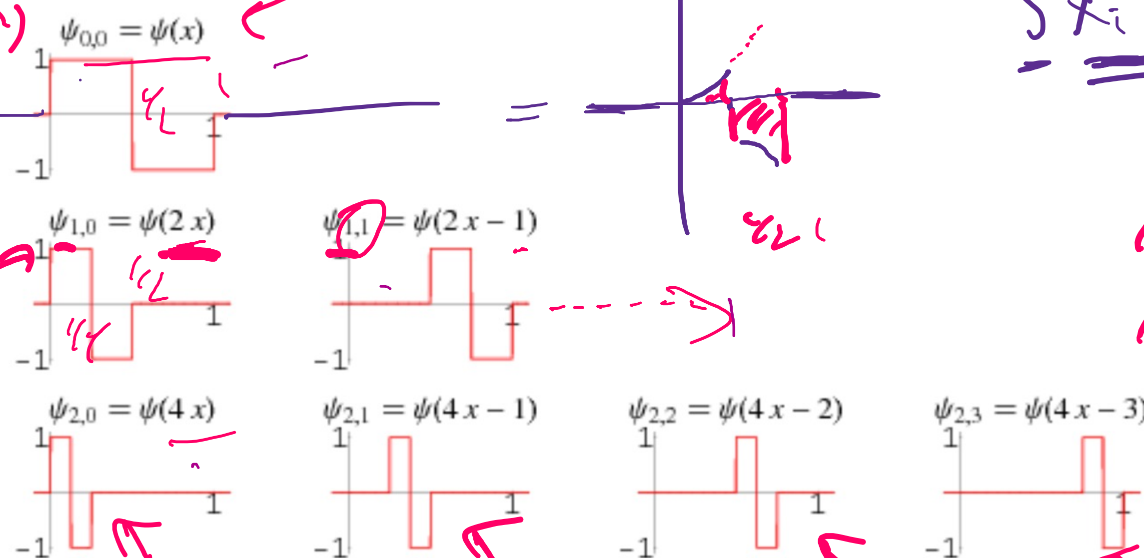
Define

$$\psi(x) \equiv \begin{cases} 1 & 0 \le x < \frac{1}{2} \\ -1 & \frac{1}{2} < x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\psi_{jk}(x) \equiv \psi(2^j x - k)$$

for j a nonnegative integer and $0 \le k \le 2^j - 1$.



Better features

$\{\phi_k\}_{k=1}^{K} = \{\chi_k\}$ indicator functions

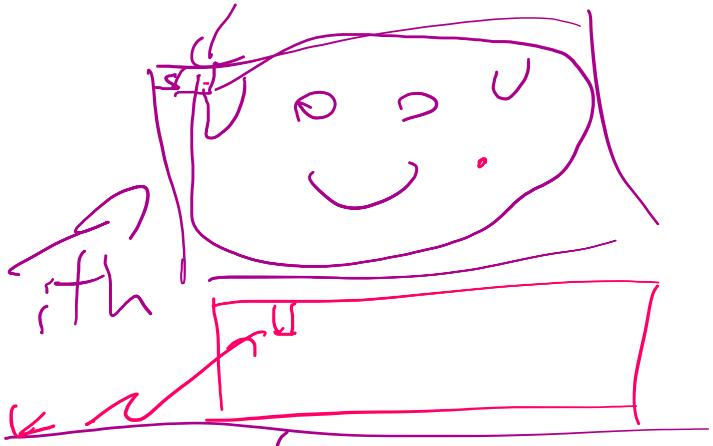
e^{iks} fourier basis ←

$\psi_k(s)$ Haar wavelet basis

ϕ_k

$e^{iks} = \cos ks + i \sin ks$

$a_k = \langle x_k, \phi_k \rangle = \int x_k(s)\phi_k(s)\,ds = \sum x_k(s)\phi_k(s)$

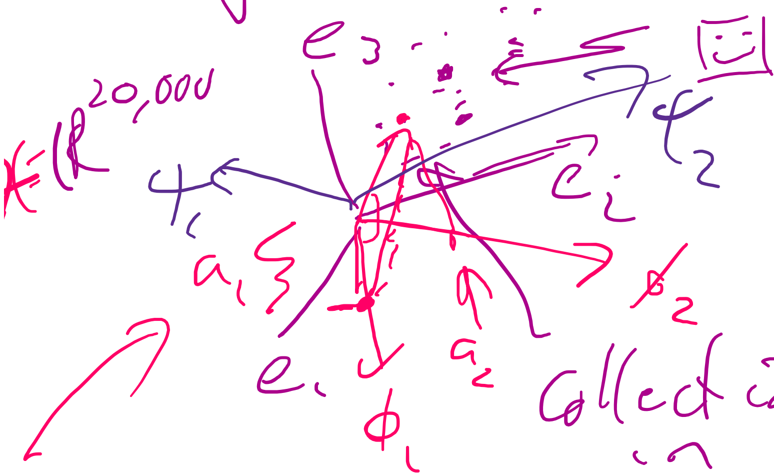


= 20,000 the values in the pixels

each pixel is considered a feature.

Project onto basis function.

Characteristic fn: $\chi_k(s) = \begin{cases} 1 & s \in k \\ 0 \end{cases}$



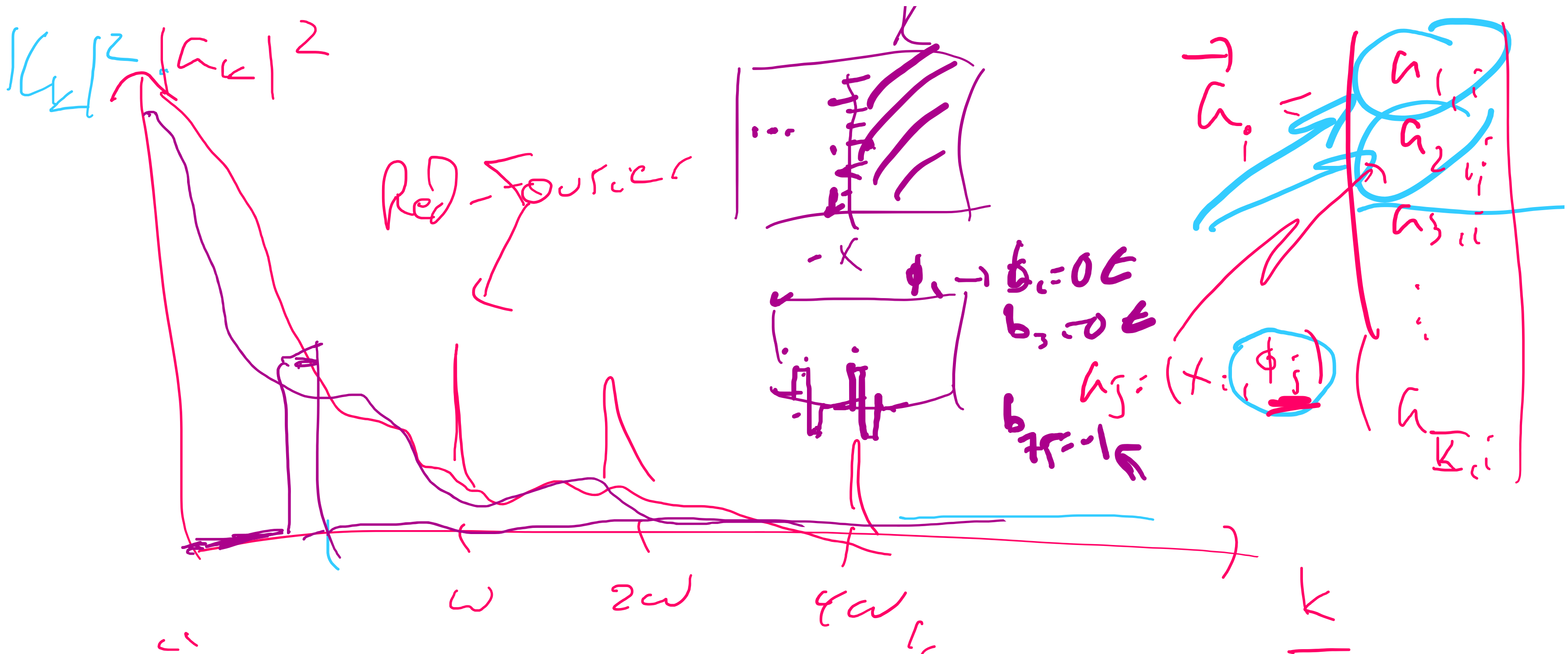
$p_k = (x_i, \chi_k) = \int x_i(s) \chi_k(s) ds$

↑ features

$\vec{p} = \begin{pmatrix} p_1 \\ \vdots \\ p_N \end{pmatrix}$

collection of pics as data cloud.

in $\mathcal{H} = \mathbb{R}^{20,000}$



"Power spectrum?"

$$\int |x(s)|^2 \, ds \overset{\le}{\simeq} \sum |a_k|^2 \longleftarrow \text{Parseval}$$

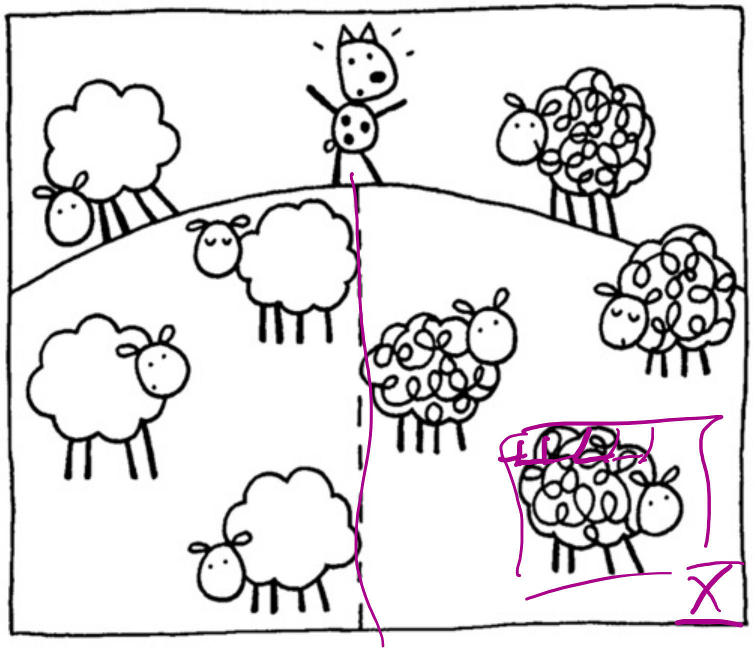
Support Vector Machines (SVM) – 5.7

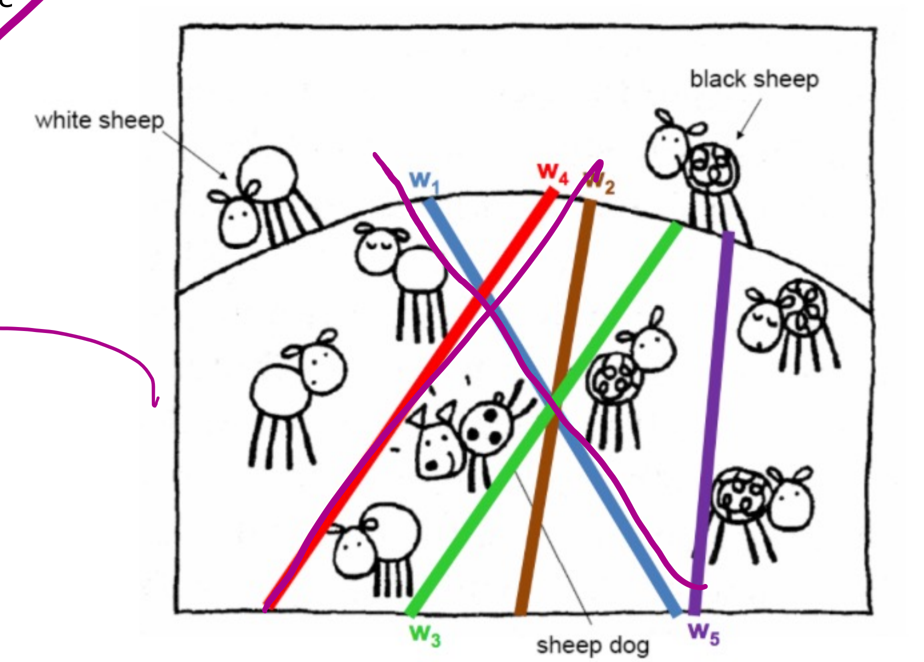
Then

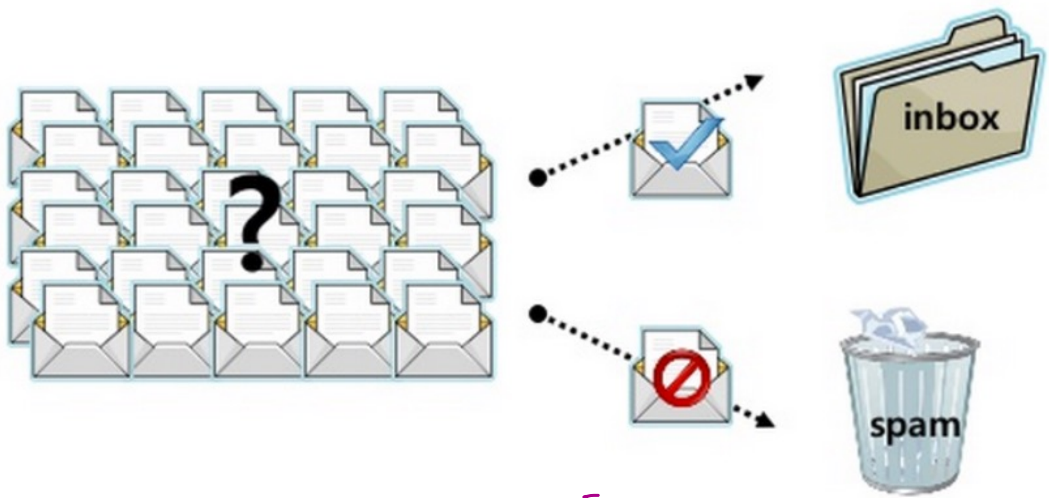
Nonlinear (kernelized) SVM (KSVM)

Wide Margin Decision Hyperplane for
Supervised - Learning Classification

First a linear binary classification – decision boundary/hyperplane







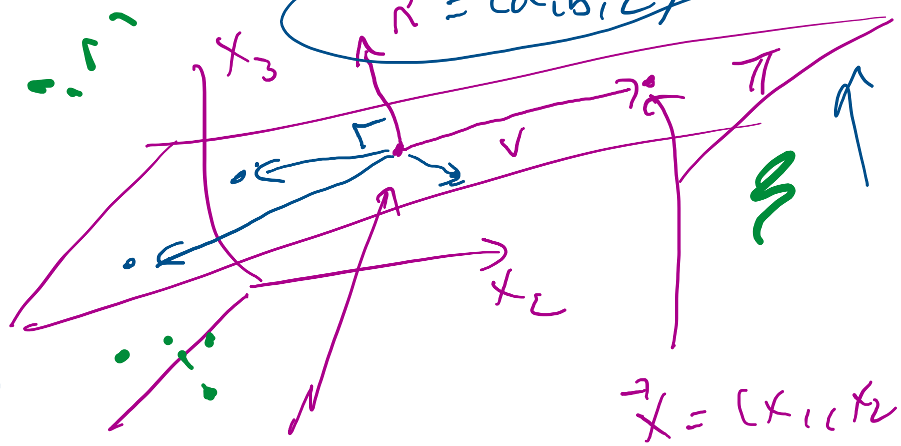
- Instance space: $x \in X$ ($|X| = n$ data points)
  - Binary or real-valued feature vector x of word occurrences
  - d features (words + other things, d~100,000+)
- Class: $y \in Y$
  - y: Spam (+1), Ham (-1)

Viagra	Learning	The	Dating	Nigeria	Is_spam
1	0	1	0	0	1
0	1	1	0	0	-1
0	0	0	0	1	1

Review – a hyperplane defined by a vector.

$\vec{n} = (a,b,c)$

- 1 eq is a co-dim 1 restriction of space



$\vec{x} = (x_1, x_2, x_3)$

$\vec{x}_0 = (x_{1,0}, x_{2,0}, x_{3,0})$

$v = \vec{x} - \vec{x}_0 = \langle (x_1 - x_{1,0}), (x_2 - x_{2,0}), (x_3 - x_{3,0}) \rangle$

$\Pi := \{ x = (x_1, x_2, x_3) : \vec{v} = \vec{x} - \vec{x}_0 \perp \vec{n} \}$

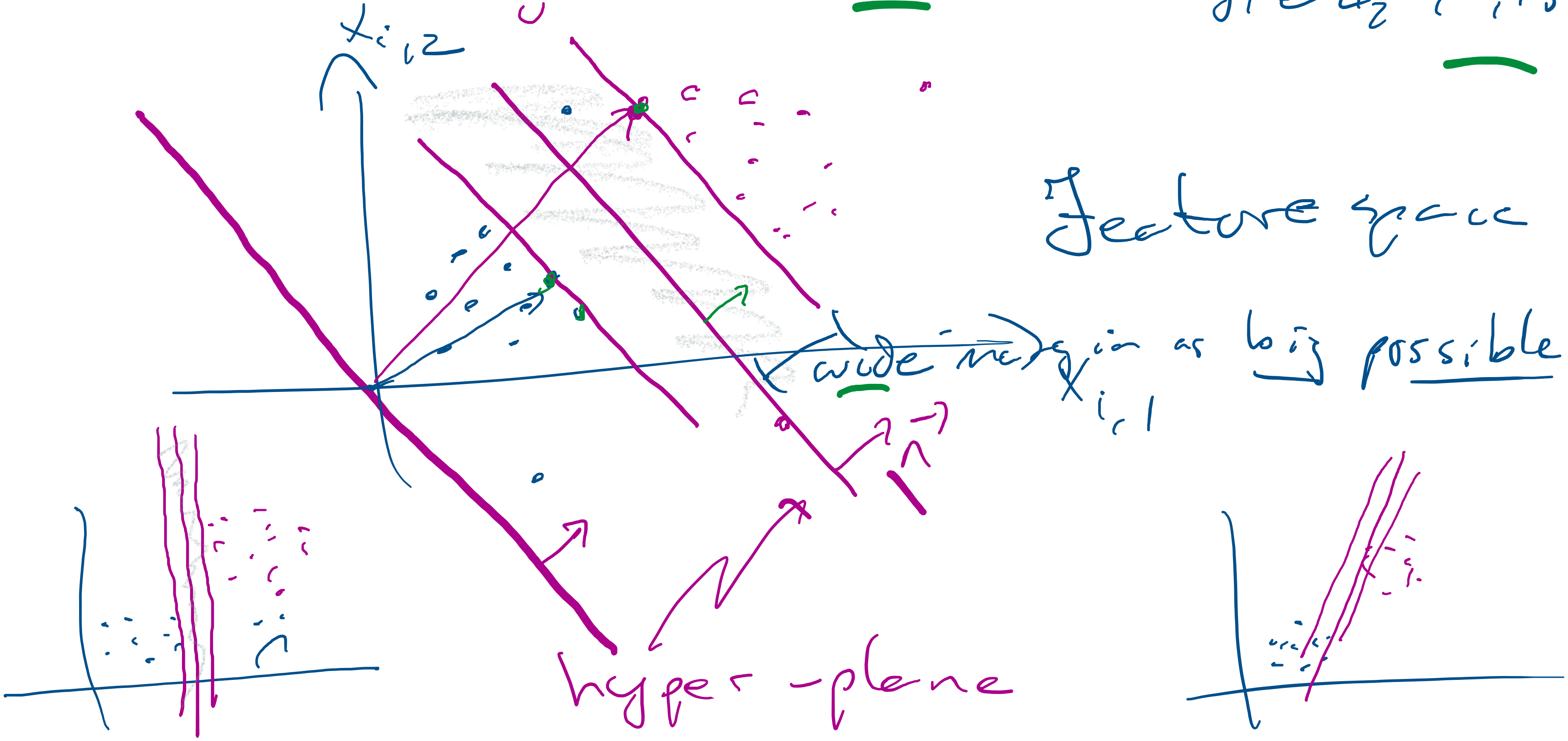
$v \cdot n = 0 \iff v \perp \vec{n}$

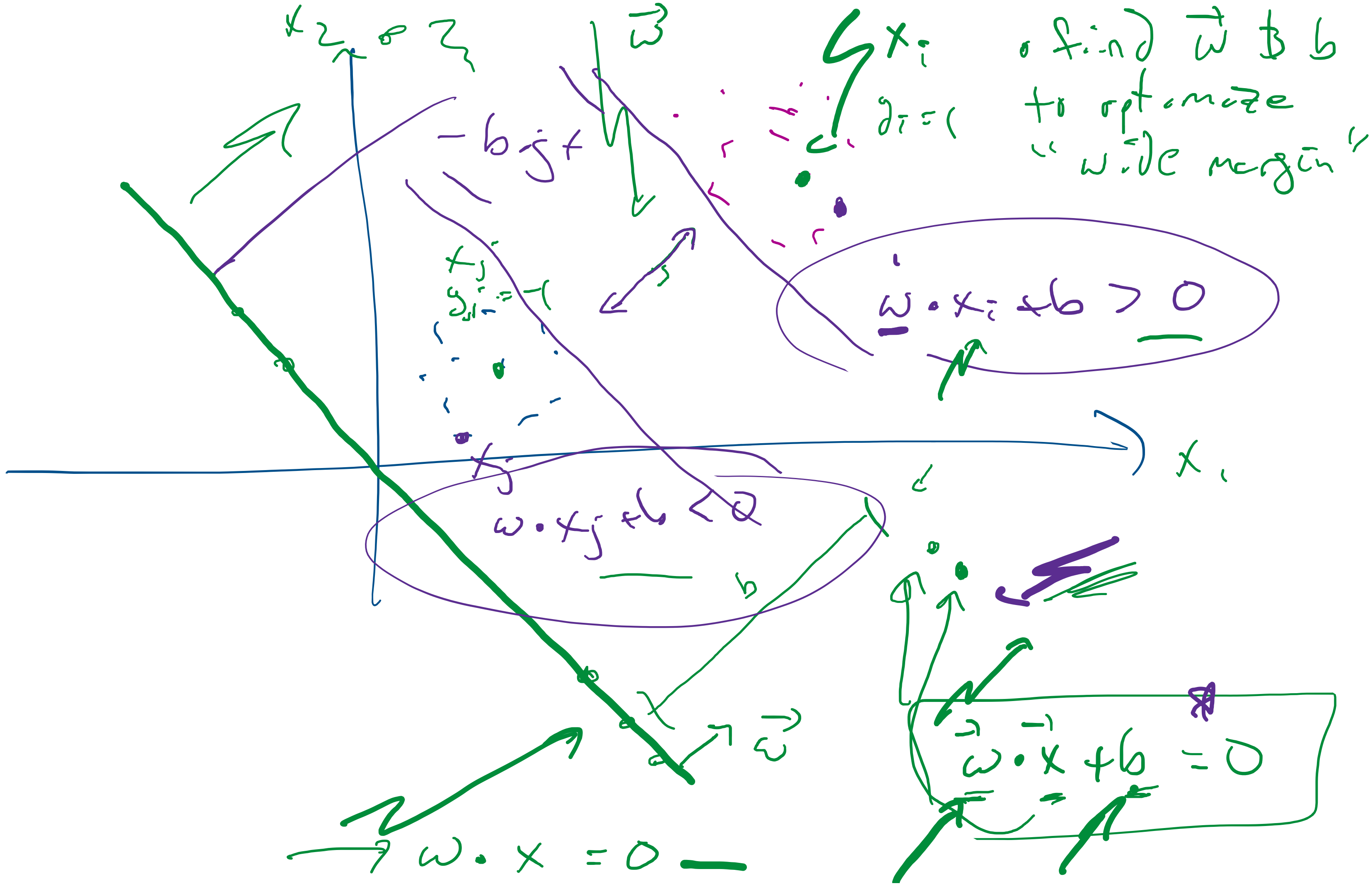
$n \cdot v = \langle a, b, c \rangle \cdot \langle x_1 - x_{1,0}, x_2 - x_{2,0}, x_3 - x_{3,0} \rangle = a(x_1 - x_{1,0}) + b(x_2 - x_{2,0}) + c(x_3 - x_{3,0}) = 0$

SVM: $D = \{(x_i, y_i)\}_{i=1}^{n}$

$x_i \in \mathbb{R}^d$

$y_i \in \mathbb{Z}_2 = \{-1, 1\}$





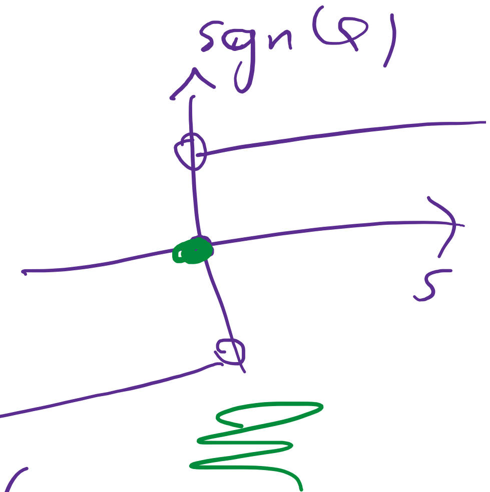
x_2
$\vec{w}$
x_i
$y_i = 1$
find $\vec{w}$ & b
to optimize
"wide margin"
$-b$
x_j
$y_j = -1$
$w \cdot x_i + b > 0$
$w \cdot x_j + b < 0$
x_1
$\vec{w} \cdot \vec{x} + b = 0$
$w \cdot x = 0$

$(\vec{x}_i, y_i)$, $y_i = -1$, 1

($y_i = 0$ or 1
apple or orange
dog or cat.)

$y_j(w \cdot x_j + b) = \text{sgn}(w \cdot x_j + b)$ good label.

$$\text{sgn}(s) = \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0 \end{cases}$$



$\text{sgn}(w \cdot x_j + b) = 1$ – labelled well –

$= -1$ mislabelled.

a loss function.

$$\ell(y_i, \bar{y}_i) = \ell(y_i, \operatorname{sgn}(w \cdot \vec{x}_i + b)) = \begin{cases} 0 & \text{if correct label } y_i = \operatorname{sgn}(w \cdot x_i + b) \\ 1 & \text{incorrect label} \end{cases}$$

$\bar{y}_i$ — label you infer from $\vec{x}_i$ alone if you have a good hyperplane $\vec{w}$ & b

Total loss

$$\sum_{i=1}^{N} \ell(y_i, \bar{y}_i)$$

$\sum$

Cost : ... ~~loss~~

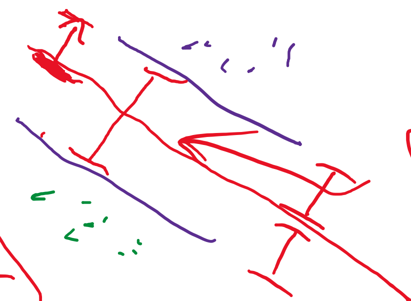
$\arg\min \frac{1}{2}\|w\|_2^2$ subj $y_i(w^T x_i - b) - 1 = 0$

$w \cdot x_j$

small $\|w\|$

subj every matches truth

dist between



dist big

$\frac{2}{\|w\|_2^2}$ big $\Leftrightarrow$ $\frac{\|w\|_2^2}{2}$ small constrained opt.

$(\vec{w}, b)$ ← $\vec{w} \in \mathbb{R}^d$, $d = 2$; $d+1 = 3$

$$\mathcal{L}(x, s, \theta) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n} s_i \left(y_i (w^T x_i - b) - 1 \right)$$

$\{(x_i, y_i)\}$

$\frac{1}{2} w \cdot w$

$y_1 (w^T x_1 - b) - 1 = 0$

$y_2 (w^T x_2 - b) - 1 = 0$

$\vdots$

$y_n (w^T x_n - b) - 1 = 0$

$$\nabla_\theta \mathcal{L} = \vec{0}$$

$$= \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n} s_i y_i (w^T \vec{x}_i - b) + \sum_{i=1}^{n} s_i$$

$$\nabla_w \mathcal{L}(x, s, \theta) = w - \sum s_i y_i \vec{x}_i = \vec{0}$$ ← d-eqns for d features

$$\nabla_b \mathcal{L} = \frac{\partial}{\partial b} \mathcal{L} = \sum s_i y_i = 0$$

$w = \sum_{i=1}^{n} s_i y_i x_i \quad ; \quad \sum_{i=1}^{n} s_i y_i = \vec{s} \cdot \vec{y} = 0$

$x_i \in \mathbb{R}^d \qquad s \in \mathbb{R}^n \qquad \in \mathbb{R}$

n-values.

"Trick" – KKT – Primal Dual form.

Primal Form. $\Big\}$ $\min_{\theta, s} \mathcal{L}(x, s, \theta) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n} s_i y_i (w^T x_i + b)$

Dual Form

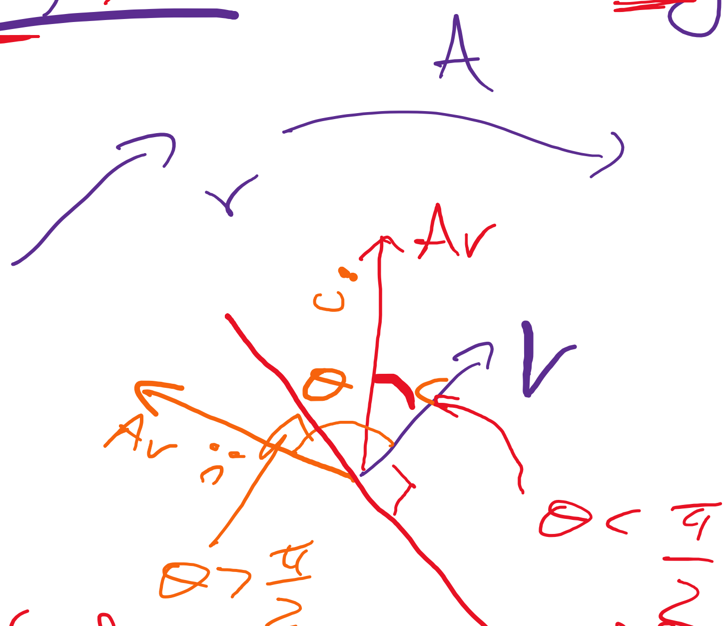
$$\begin{cases} \max \; \mathcal{L}_D(x, s, \theta) = \sum s_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} s_i s_j y_i y_j (x_i \cdot x_j) \\ \text{s.t.} \quad w = \sum_{i=1}^{n} s_i y_i x_j \text{ and} \\ \sum_{i=1}^{n} s_i y_i = 0 \end{cases}$$

$K(x_i, x_j)$

$\phi(x_i) \cdot \phi(x_j)$

Pos. Semi-Defn Definition -

- A matrix $A_{n \times n}$ is (semi) positive definite if $v \cdot (A \cdot v) \geq 0$ for any $v \neq 0$ in domain of A.



$$v \cdot (Av) = \|v\| \, \|Av\| \cos\theta$$

- A kernel fn is pos. semi definite if $x_1, x_2, x_3,$ and $\bar{K}$ pos. semi def matrix for any input set

- $\mathcal{H}$ = Hilbert space – a complete inner product space

- a Hilbert space is a set $\mathcal{H}$ of vectors such that there is a complete inner product.

vector space

complete → limits work properly.

inner product → dot

Spectral Decomp. thm:

Suppose $A_{n \times n}$ is pos. defn. symm. matrix with eigenvectors & eigenvalues λ_i, v_i, $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

not $v_i^T v_i = v_i \cdot v_i$

$v_i \otimes v_i = P_i$ rank-1 projector

$\swarrow \underline{X} = \mathbb{R}^2$

- $x_i \cdot x_j$ need the dot product between $\underline{x}_i$ & $\underline{x}_j$ to do SVM.

- $\underline{K(x_i, x_j)} = \phi(x_i) \cdot \phi(x_j)$

  $\uparrow$ corresponds. $\uparrow$

  $\phi : \underline{X} \to \mathcal{H}$ $\swarrow \mathbb{R}^6$

Gram matrix - symmetric

$$\underline{K} = \begin{pmatrix} \underline{K(x_1, x_2)} & K(x_1, x_3) & \cdots & \underline{K(x_1, x_n)} \\ \vdots & & & \\ \underline{K(x_n, x_1)} & \cdots & & K(x_n, x_n) \end{pmatrix}$$

KKT

Primal Problem:

$$\begin{cases} \text{minimize: } \mathcal{L}(x,s) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} s_i y_i (w^T x_i + b) + \sum_{i=1}^{n} s_i \\ \text{such that: } s_i \geq 0, \forall i \end{cases}$$

Dual Problem:

$$\begin{cases} \text{maximize: } \mathcal{L}_D(x,s) = \sum_{i=1}^{n} s_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j y_i y_j (\vec{x}_i^T \vec{x}_j) \\ \text{using: } w = \sum_{i=1}^{n} s_i y_i x_i, \text{ and } \sum_{i=1}^{n} s_i y_i = 0 \end{cases}$$

The amazing Kernel trick – nonlinear SVM through a kernel and all dot products in the high dimensional space
Done through a kerekl function

Now, we define a kernel $K : X \times X \mapsto \mathbb{R}$, which can take different forms such as:

- Linear kernel: $K(x, \tilde{x}) = x^T \tilde{x}$.
- Polynomial kernel: $K(x, \tilde{x}) = (x^T \tilde{x} - y)^d$.
- Gaussian RBF: $K(x, \tilde{x}) = e^{-\frac{Vert x - \tilde{x}\|^2}{2\sigma^2}}$

Consider the polynomial kernel, for $d = 2$, $X = \mathbb{R}^2$, then we have:

$$
\begin{aligned}
K(x, \tilde{x}) &= (x \cdot \tilde{x} + 1)^d \\
&= (x^T \tilde{x} + 1)^d \\
&= (x_1\tilde{x}_1 + x_2\tilde{x}_2 + 1)^2 \\
&= x_1^2\tilde{x}_1^2 + 2x_1\tilde{x}_1 + 2x_2\tilde{x}_2 + x_1\tilde{x}_1x_2\tilde{x}_2 + 1
\end{aligned}
$$

which interestingly can be re-written in terms of dot product:

$$
\begin{aligned}
K(x,\tilde{x}) &= (x \cdot \tilde{x} + 1)^d \\
&= (x^T \tilde{x} + 1)^d \\
&= (x_1\tilde{x}_1 + x_2\tilde{x}_2 + 1)^2 \\
&= x_1^2\tilde{x}_1^2 + 2x_1\tilde{x}_1 + 2x_2\tilde{x}_2 + x_1\tilde{x}_1x_2\tilde{x}_2 + 1
\end{aligned}
$$

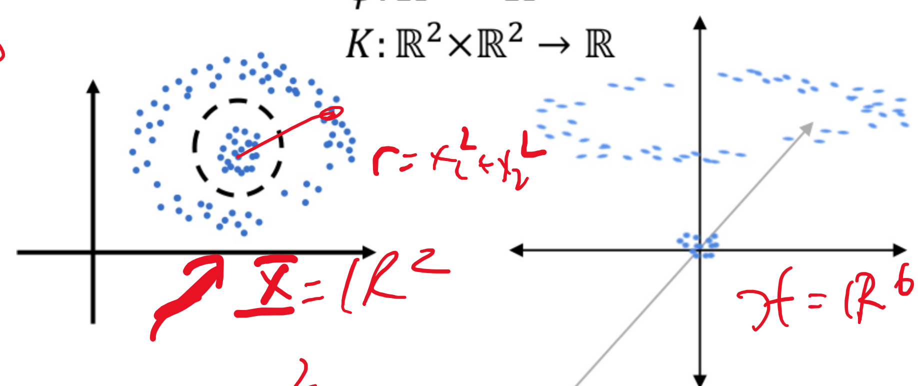
which interestingly can be re-written in terms of dot product:

$$K(x,\tilde{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_1x_2, x_2^2)\cdot(1, \sqrt{2}\tilde{x}_1, \sqrt{2}\tilde{x}_2, \tilde{x}_1^2, \tilde{x}_1\tilde{x}_2, \tilde{x}_2^2)$$

$$\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^6$$

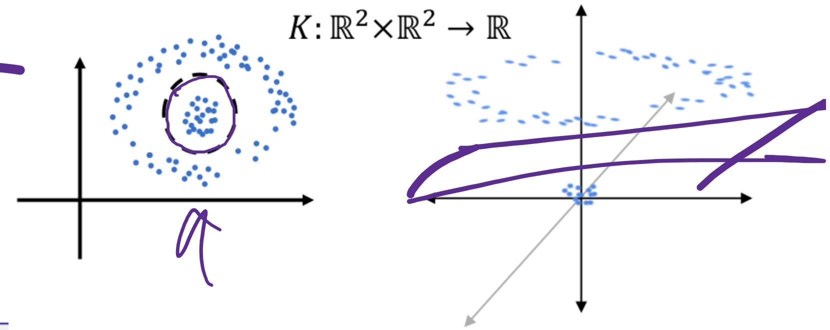
$$K: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$K(x,\tilde{x}) = (\phi(x) \cdot \phi(\tilde{x}))$$



$$\phi(x_1, x_2) = (\phi_1(x_1, x_2), \phi_2(x_1, x_2), ..., \phi_6(x_1, x_2))$$

where $\phi : X \mapsto \mathcal{H}$.



Note that $X = \mathbb{R}^2$ is the domain, and $\mathcal{H}$ is the Hilbert space, which is (in machine learning literature) the feature space, and a set of features $\phi_i, \forall i$, is called dictionary.

Mantra

A major theme in machine learning is that sometimes things actually get easier in higher dimensions !!!.

- A linear plane in high dimensional feature space $\mathcal{H}$, may be a nonlinear curves in the domain space.
- $\mathcal{H}$ is a plane, with calculus with dot products is legit.

The following, we introduce Mercer's theorem, which generalizes spectral decomposition theorem.

Theorem 5.5.1 — Mercer's Theorem. Let $K \in L^2(X \times X)$, (i.e. $\int |K(x,\tilde{x})|^2 dx d\tilde{x} < \infty$) such that $T : L_2(X) \mapsto L_2(X)$ by $(T(f)(x)) = \int K(x,\tilde{x}) f(\tilde{x}) d\tilde{x}$ is positive definite. If $\phi_i \in L^2(X)$ is a normalized eigenfunction with eigenvalues $0 < \lambda_1, \leq \lambda_2 \leq \cdots \leq \lambda_N$, Then

$$K(x,\tilde{x}) = \sum_{i=1}^{N_{\mathcal{H}}} \lambda_i \phi_i(x) \phi_i(\tilde{x}) \tag{5.20}$$

for almost every $(x,\tilde{x})$. Where $N_{\mathcal{H}} = dim(\mathcal{H})$, and the convergance of $K(x,\tilde{x})$ is absolute.

Mercer's theorem itself is a generalization of the result that any symmetric positive-semidefinite matrix is the Gramian matrix of a set of vectors.

$$D = \{(x_i, y_i)\}$$