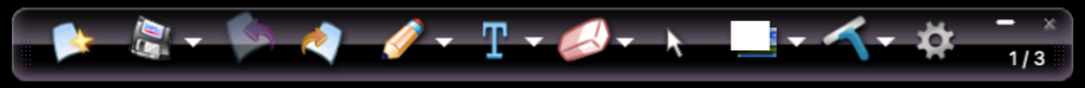
Topics:

- GMM - optimization ✓
- GMM: EM algorithm ✓

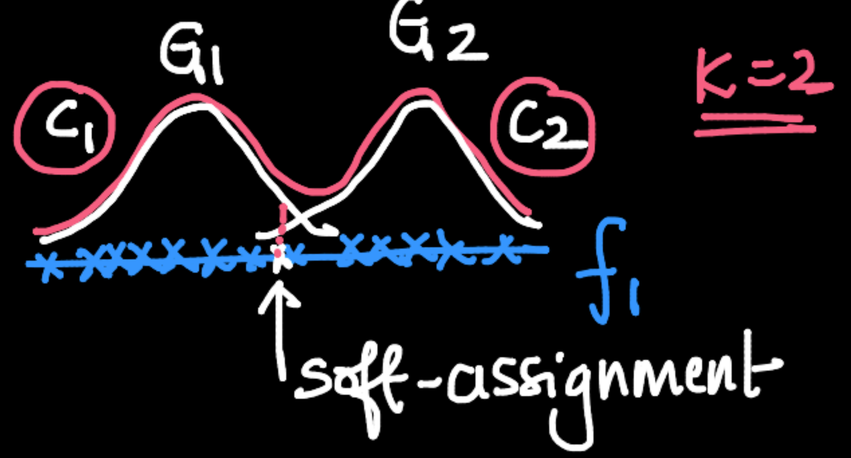
assumption → ✓

SVM → Kernel tricks

- Misc. topics - 1
  - ✓ Lasso, Ridge & ElasticNet (L2)
  - ✓ Calibration: Platts-scaling; Isotonic Regression
  - K-Means++ (Speedup)
  - Robust models: RANSAC



Recap of GMM: → Math-heavy



1. Mixture-model →
2. 1D-GMM
3. Multi-dimensional Gaussian : $N_d(\mu_{d1}, \Sigma_{d\times d})$
   - ↳ cov-matrix
   - ↳ bell in d+1-dim ← PDF

1D Gaussian: $N(\mu, \sigma)$

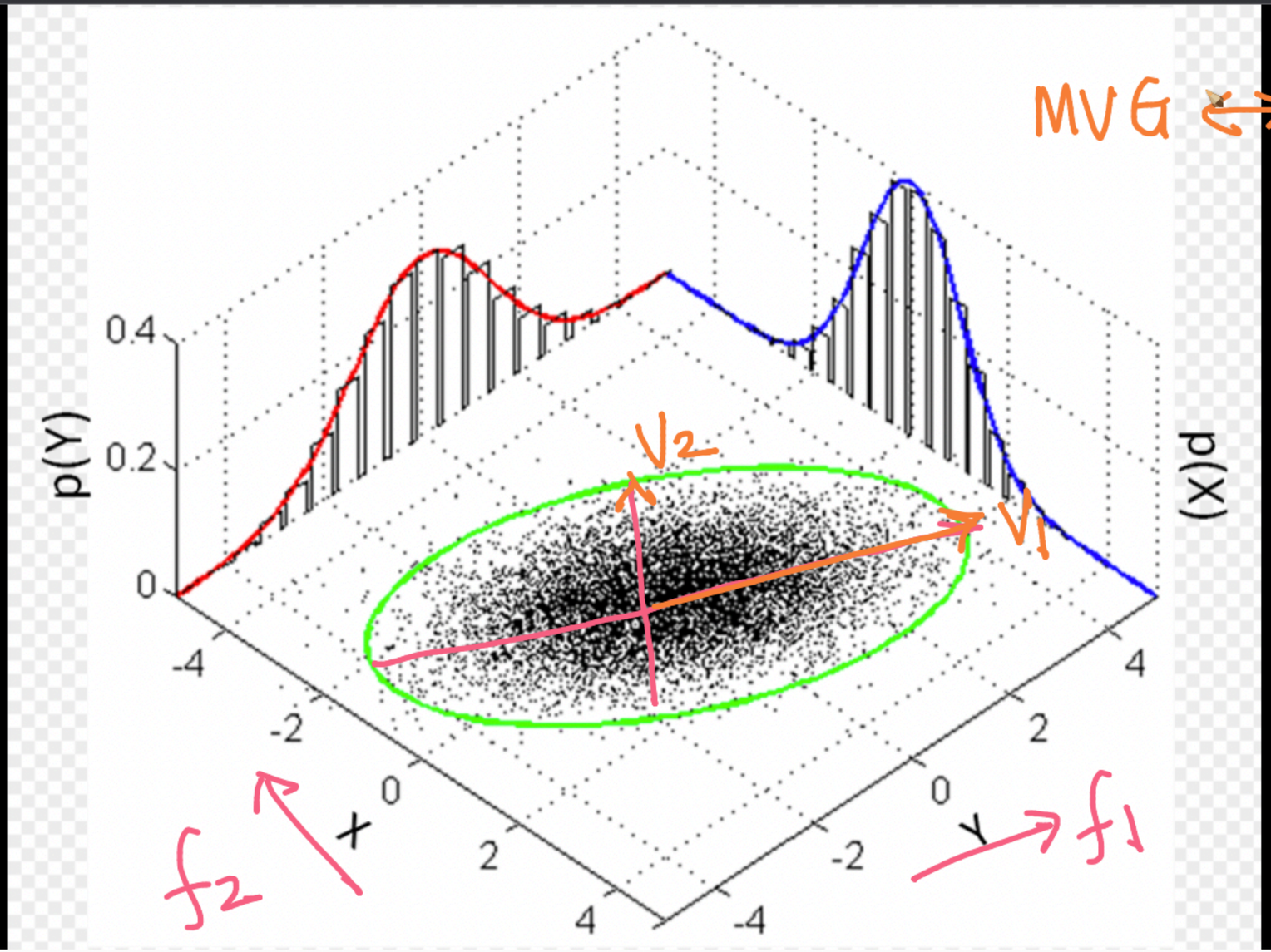


Illustration of a multivariate gaussian distribution and its marginals. Matlab code provided below.

$x_1, x_2 \dots x_n$

1D - Gaussian:

$X \sim N(\mu, \sigma)$

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

d-dim Gaussian:

dxd symm - Matrix

params

$$\mathcal{D}: \underbrace{x_1, x_2 \dots x_i \dots x_n}_{} : X \sim N_d(\mu_d, \Sigma_{dxd})$$

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

$$\mathcal{D} = \begin{array}{c} f_1\ f_2 \dots f_d \\ \hline x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{array}$$

$$\begin{cases} \hat{\mu}_d = \frac{1}{n}\sum_{i=1}^{n} x_i \\ \hat{\Sigma}_{ij} = \text{cov}(f_i, f_j) \end{cases}$$

$\hat{\mu}_d$: d-dim vec

x_i: d-dim vec

$\hat{\Sigma}$: ↳ PCA (eavlier)

Chrome File Edit View History Bookmarks Profiles Tab Window Help 9:19

Find k- Gaussians → $\mu_d^j, \Sigma_{d\times d}^j \quad \forall j = 1 \to k$

Given: $\mathcal{D} = \{x_i\}_{i=1}^{n} \quad x_i \in \mathbb{R}^d$

← { y_j: index of the Gaussian

$j : 1 \to k$

6 / 7

Generative Methods (GMM)

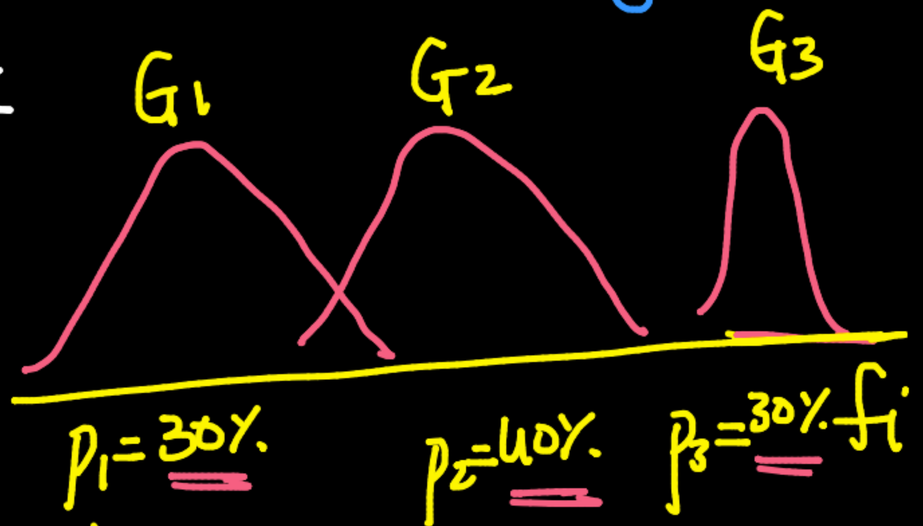
Assumptions on how the data was generated

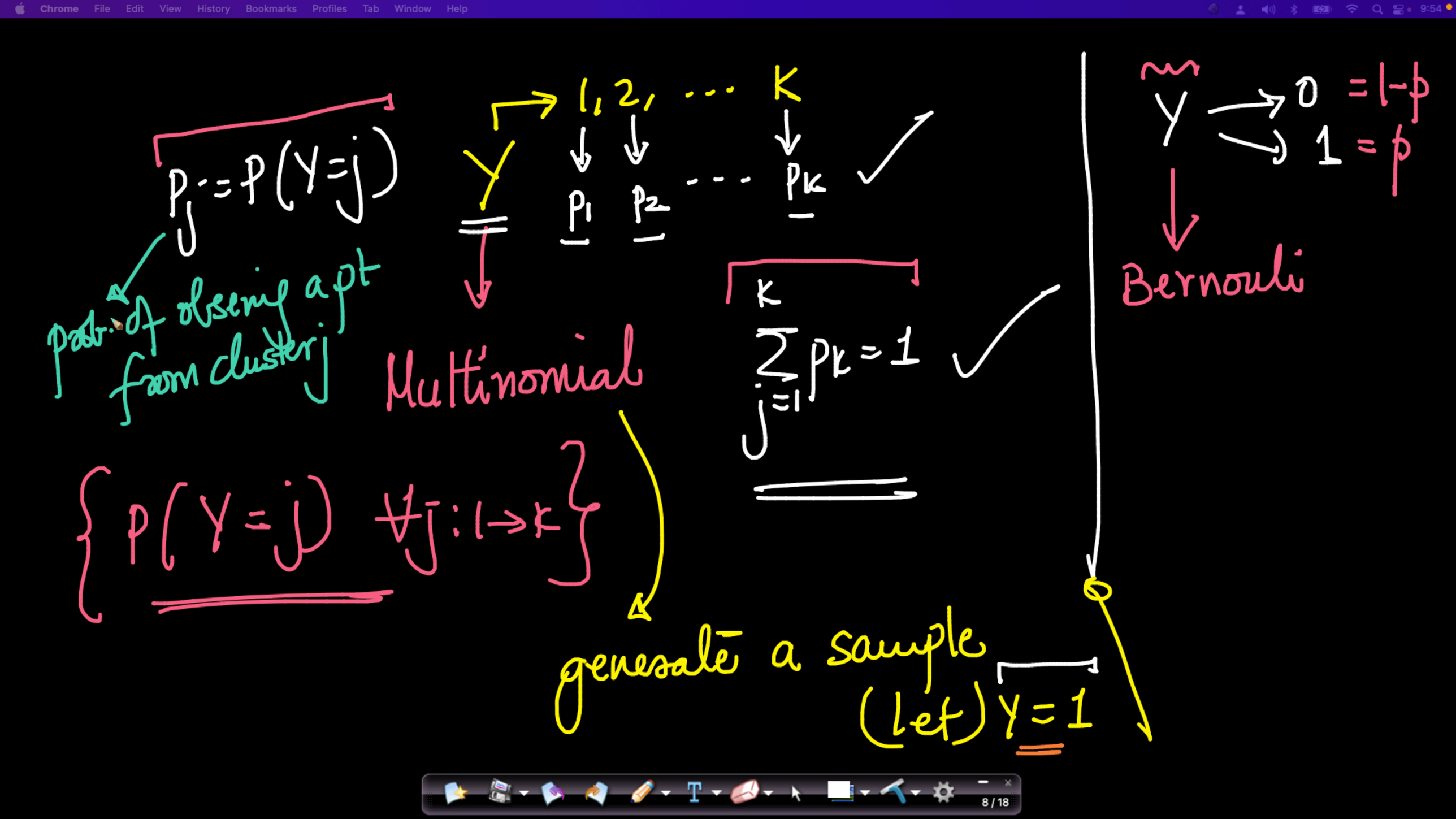
Generative process underlying GMM

(a) → (*) $P(Y = j) \; \forall j : 1 \to k$

$$\begin{cases} P(Y=1) = 0.3 \\ P(Y=2) = 0.4 \\ P(Y=3) = 0.3 \end{cases}$$

K-gaussians




$p_j := P(Y=j)$
prob. of observing a pt from cluster j
$Y \to 1, 2, \dots K$
$p_1 \; p_2 \; \dots \; p_K$
$\sum_{j=1}^{K} p_k = 1$
Multinomial
$\{ P(Y=j) \; \forall j : 1 \to K \}$
generate a sample
(let) $Y = 1$
$Y \to 0 = 1-p$
$1 = p$
Bernoulli

(6)

(+) μ^j_{d}, $\Sigma^j_{d\times d}$ $\forall j = 1 \to K$

↳ params of each underlying Gaussian

$\because y = 1$

Sample from $N_d(\mu'_d, \Sigma'_{d\times d})$

↳ one data point

Repeat steps a & b many times

↓

you can obtain data very "similar" to observed D

Discriminative-methods

all techniques fall now

$x_i \rightarrow$ Cl0 or Cl 1

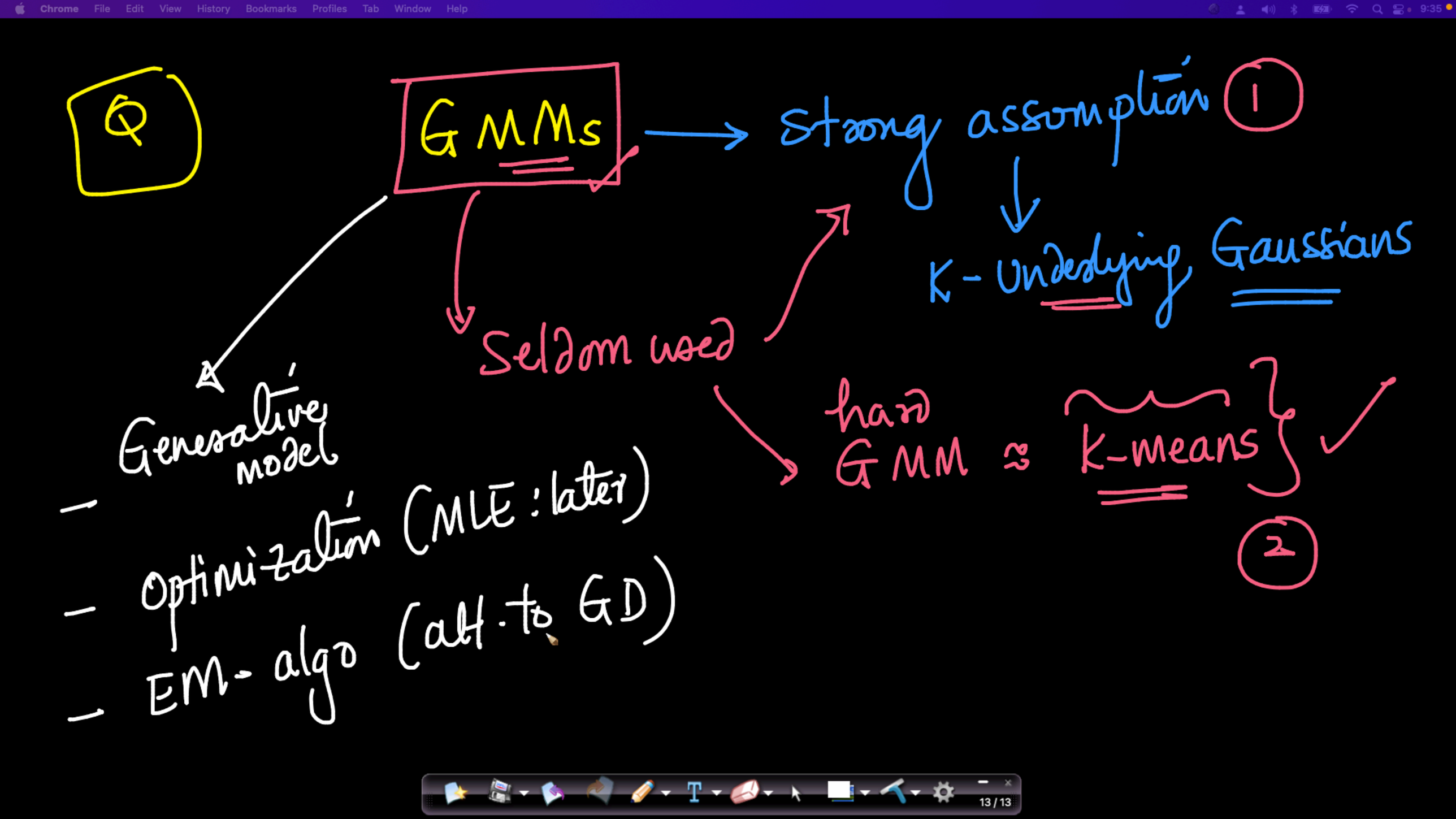
$x_i \rightarrow$ cluster 1 or cluster 2

$$\{ \quad p(Y=j) = p_j \qquad \sum_{j=1}^{k} p_j = 1$$

$$1, 2, \cdots j \cdots k$$

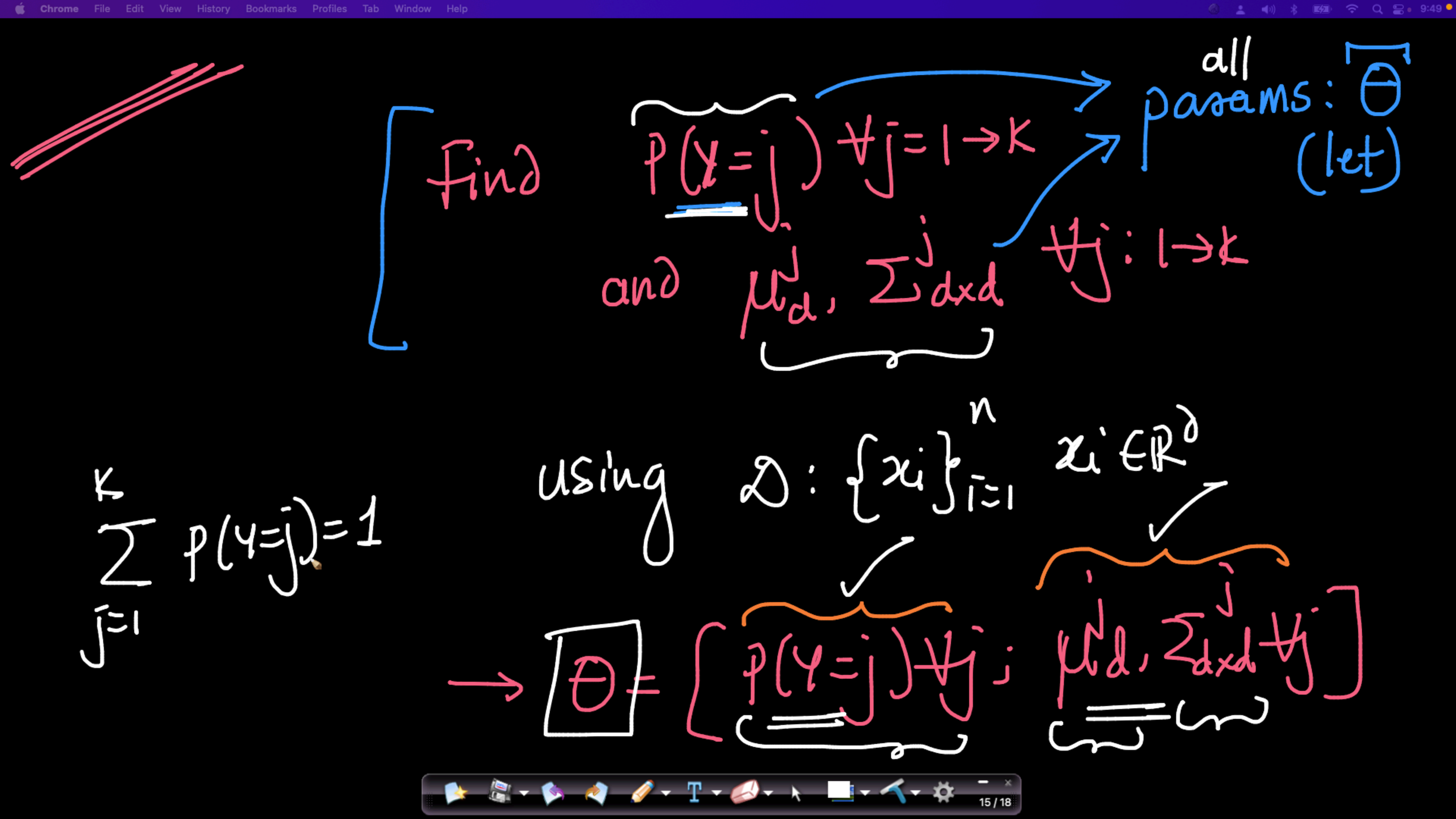
① $P(Y_i = j \mid x_i)$

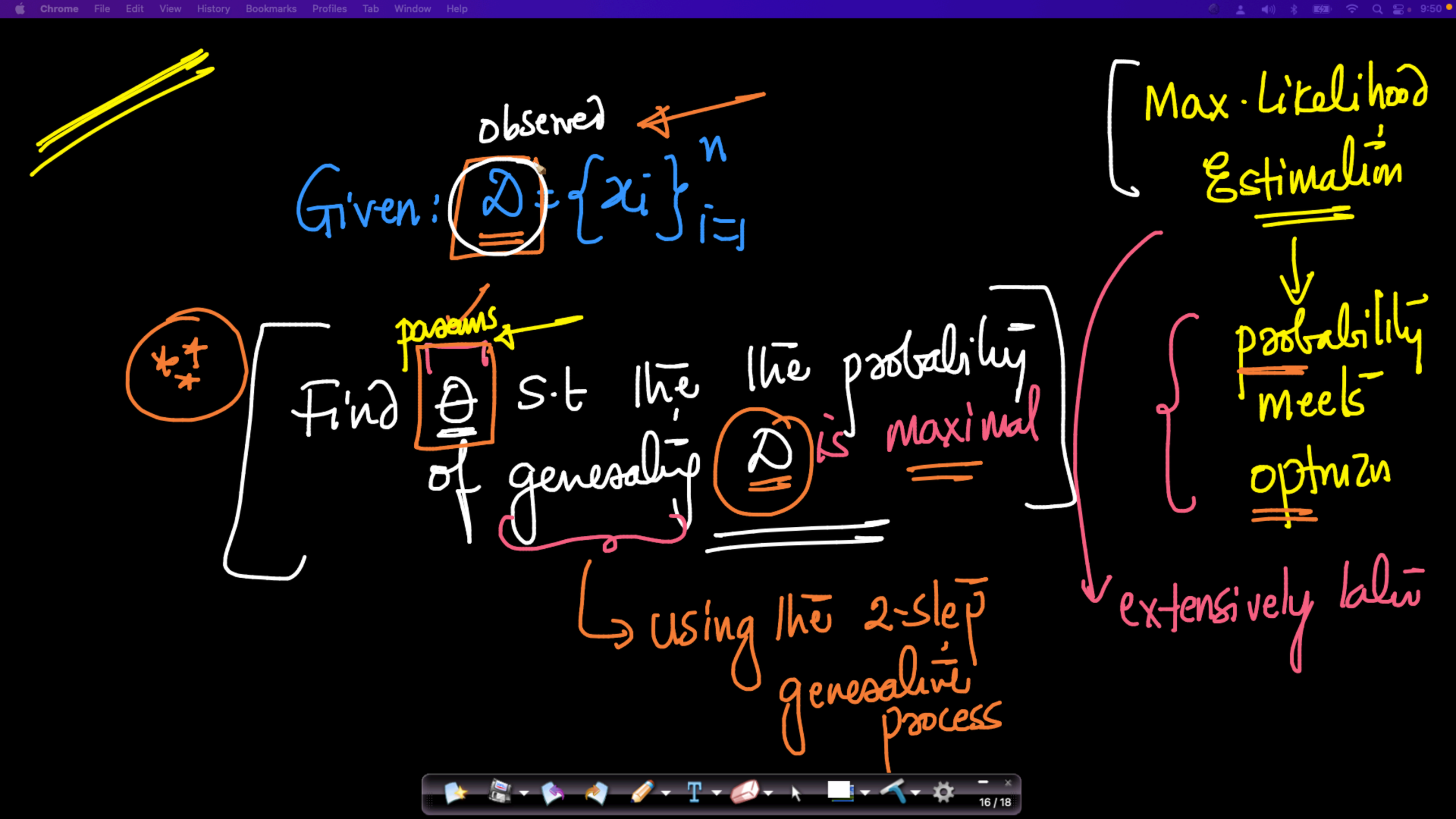
② $\sum_{j=1}^{k} P(Y_i = j \mid x_i) = 1$


Q
GMMs
Strong assumption (1)
K - Underlying Gaussians
Seldom used
Generative model
- Optimization (MLE : later)
- EM - algo (alt. to GD)
hard GMM ≈ K-means
(2)

Recap:

- Compute μ_d, $\Sigma_{d \times d}$ given D: $x_i \in \mathbb{R}^d$
- Generative process underlying GMM (assumption)


find $P(Y=j)\ \forall j=1\to K$
all params: θ (let)
and $\mu_d^j, \Sigma_{d\times d}^j$
$\forall j: 1\to K$
$\sum_{j=1}^{K} P(Y=j)=1$
using $\mathcal{D}: \{x_i\}_{i=1}^{n}$
$x_i \in \mathbb{R}^d$
$\theta = [P(Y=j)\ \forall j;\ \mu_d^j, \Sigma_{d\times d}^j\ \forall j]$
15 / 18


Chrome File Edit View History Bookmarks Profiles Tab Window Help
observed
Given: $\mathcal{D} = \{x_i\}_{i=1}^{n}$
Find θ s.t the the probability of generating $\mathcal{D}$ is maximal
params
using the 2-step generative process
Max. Likelihood Estimation
probability meets optmzn
extensively talk
16 / 18

hard GMM ≈ k-means (discriminative)

↑ generative method

GMM ≈ soft k-means

↑

$$\max_{\theta} \; P(D)$$

{ [generative process underlying GMM]

$$\Rightarrow \max_{\theta} \; P(x_1, x_2, \ldots x_n)$$

$$D = \{x_i\}_{i=1}^{n}$$

each x_i is indep of one another

$$\Rightarrow \max_{\theta} \; \prod_{i=1}^{n} P(x_i)$$

↳ prob. of generating x_i

$$\max_{\theta} \prod_{i=1}^{n} \underbrace{P(x_i)}_{\text{prob of observing } x_i}$$

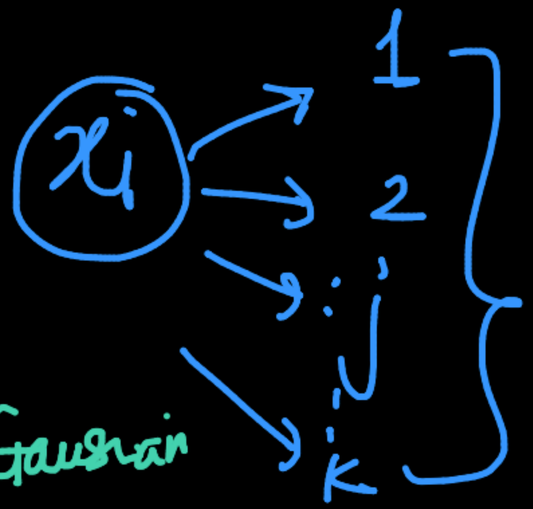
Generalized process:

$$= \max_{\theta} \prod_{i=1}^{n} \sum_{j=1}^{K} P(y_j, x_i)$$

(a) one j using P_j's

(b) generate data from $N(\mu_d^j, \Sigma_d^j)$ and

Prob of observing x_i being generated by j^{th} Gaussian



$$= \max_{\theta} \prod_{i=1}^{n} \sum_{j=1}^{k} P(y_j, x_i) \rightarrow P(y_j \cap x_i)$$

conditional - post

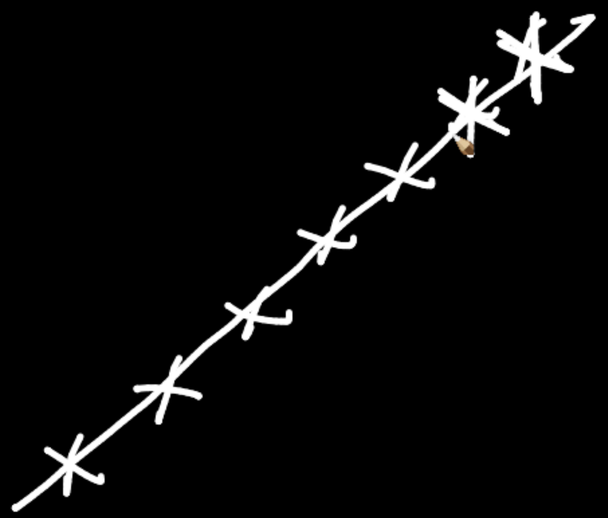
$$= \max_{\theta} \prod_{i=1}^{n} \sum_{j=1}^{k} \left(P(x_i | y_j) \; P(y_j) \right)$$

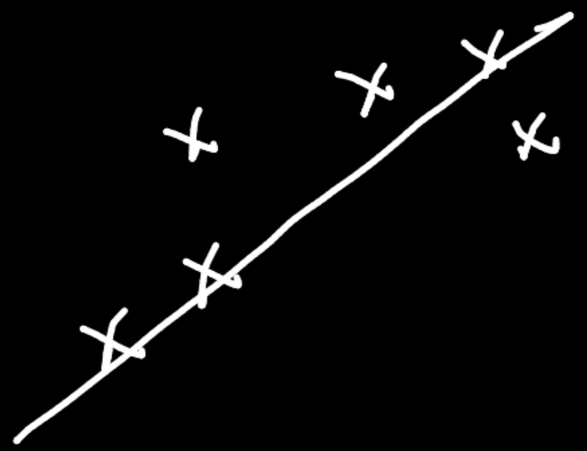
↳ class prior

Params $\theta = \left[P(Y=j); \; \mu_{d}^{j}, \Sigma_{d \times d}^{j} \; ; j : 1 \to k \right]$

likelihood of generating x_i using j^{th} Gaussian

Q





$$\max_{\theta} \prod_{i=1}^{n} \sum_{j=1}^{K} P(x_i \mid y_j) \cdot P(y_j)$$

$\theta = (P(y=j) ; \mu^j, \Sigma^j)$

$p_j = P(y_j)$

plug in the PDF of $N(\mu_d^j, \Sigma_{d \times d}^j)$

PMF of a multinomial r.v

The spherical normal distribution can be characterised as the unique distribution where components are independent in any orthogonal coordinate system.[3][4]

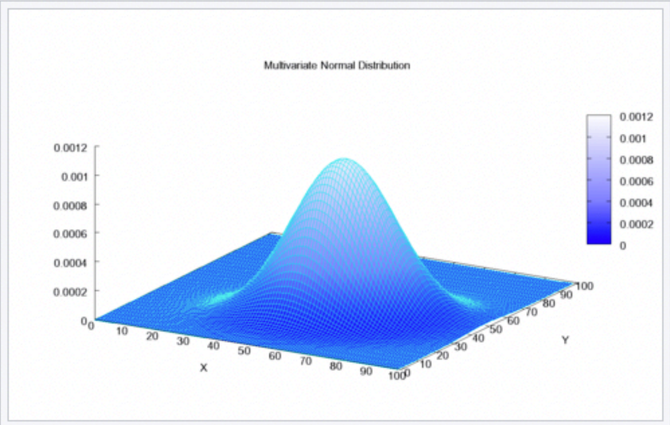
Density function [edit]

Non-degenerate case [edit]

The multivariate normal distribution is said to be "non-degenerate" when the symmetric covariance matrix $\boldsymbol{\Sigma}$ is positive definite. In this case the distribution has density[5]

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

where $\mathbf{x}$ is a real k-dimensional column vector and $|\boldsymbol{\Sigma}| \equiv \det \boldsymbol{\Sigma}$ is the determinant of $\boldsymbol{\Sigma}$, also known as the generalized variance. The equation above reduces to that of the univariate normal distribution if $\boldsymbol{\Sigma}$ is a 1×1 matrix (i.e. a single real number).



Bivariate normal joint density

The circularly symmetric version of the complex normal distribution has a slightly different form.

Each iso-density locus — the locus of points in k-dimensional space each of which gives the same particular value of the density — is an ellipse or its higher-dimensional generalization; hence the multivariate normal is a special case of the elliptical distributions.

The quantity $\sqrt{(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$ is known as the Mahalanobis distance, which represents the distance of the test point $\mathbf{x}$ from the mean $\boldsymbol{\mu}$. Note that in the case when $k = 1$, the distribution reduces to a univariate normal distribution and the Mahalanobis distance reduces to the absolute value of the standard score. See also Interval below.

Bivariate case [edit]

In the 2-dimensional nonsingular case ($k = \operatorname{rank}(\Sigma) = 2$), the probability density function of a vector $[\mathrm{XY}]'$ is:

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{\quad}}\left(\quad\right)\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 \quad \frac{(x-\mu_X)(y-\mu_Y)}{\quad} \quad \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\Bigg)$$

$$\max_{\theta} \prod_{i=1}^{n} \sum_{j=1}^{k} P(x_i \mid y_j) \cdot P(y_j)$$

$P(y_j)$ → one of the params

$P(x_i \mid y_j)$ → PDF of a MDG

μ_d^j, Σ^j $d \times d$

Σ^{-1}, $|\Sigma|$ ...

Complex →

- complex to solve ✓
- GD - complex
- multiple local minima

'Hack' to solve the optmzn w/o using GD ✓

↳ Expectation - Maximization :

Cool-idea!

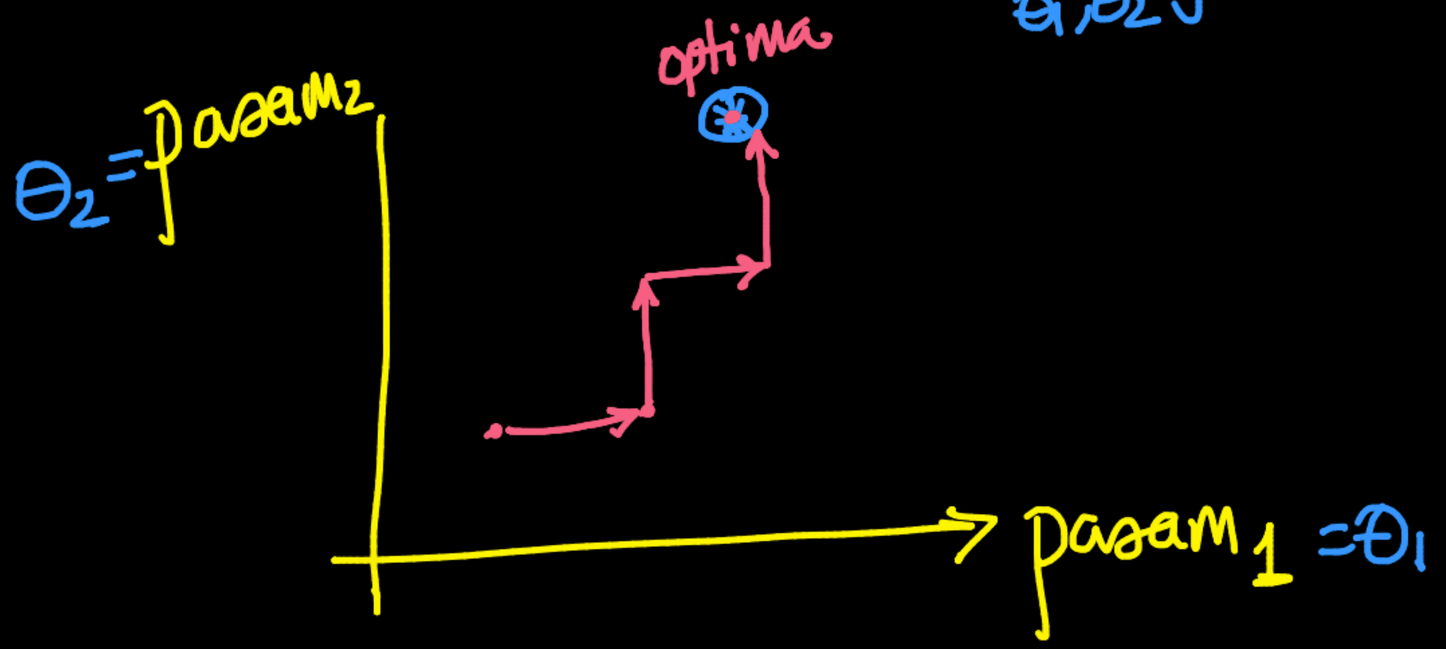
$$\theta = \left[\underbrace{\pi_j}_{K},\ \underbrace{\mu_{d}^{j}}_{kxd},\ \underbrace{\Sigma_{dxd}^{j}}_{kxdxd}\ \forall j = 1 \to K \right]$$

→ { E: update ; ✓ — fix
M: fix ; — update ✓

→ Computationally much cheaper than GD

EM: coordinate ascent

$\max_{\theta_1, \theta_2} f(\theta_1, \theta_2)$



Recap:

→ Gaussian Mixture-Model (n-dim) → θ → P_j ; μ_d^j ; Σ_{dxd}^j

$j : 1 \to k$

→ Generative process → P_j

μ_d^j, Σ_{dxd}^j

→ $\max_{\theta} P(\mathcal{D})$

break: 10:35

GMM

Given: x_i's $i = 1 \to n$; μ^j, Σ^j (very similar K-means)

1. Exp: (soft assignment)

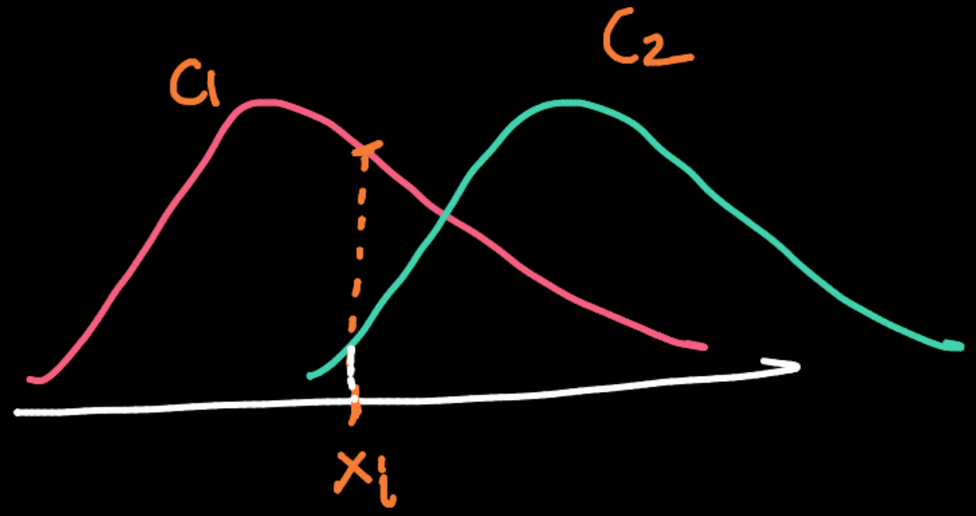
for each x_i, we compute the prob it belongs to j^{th} cluster (PDF)

$P(x_i \in j^{th})$ using PDF + normalize

2. Max: (update)

Re-estimate Gaussian-params for all Gaussians

update μ^j & Σ^j using the weighted scheme



$P(x_i \in C_1)$: PDF = p_1

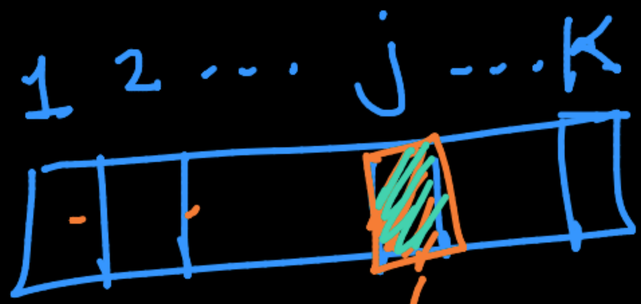
$P(x_i \in C_2)$: PDF = p_2

$\frac{p_1}{p_1+p_2}$

$\frac{p_2}{p_1+p_2}$

@ end of exp-step :-

$\forall x_i$:



→ sum to 1

↳ $p(x_i \in C_j)$

✓ In Max-step :

μ^j = weighted average of each x_i

Same as Centroid

↳ $P(x_i \in C_j)$

Σ^j : weighted Co-var

soft / fuzzy K-means (optmzn)

→ very similar to GMM

makes no assumption

assume Gaussians underlying

× GMM: → soft-clustering...

→ less used in practice

∴ hard GMM ≡ K-means (popular)

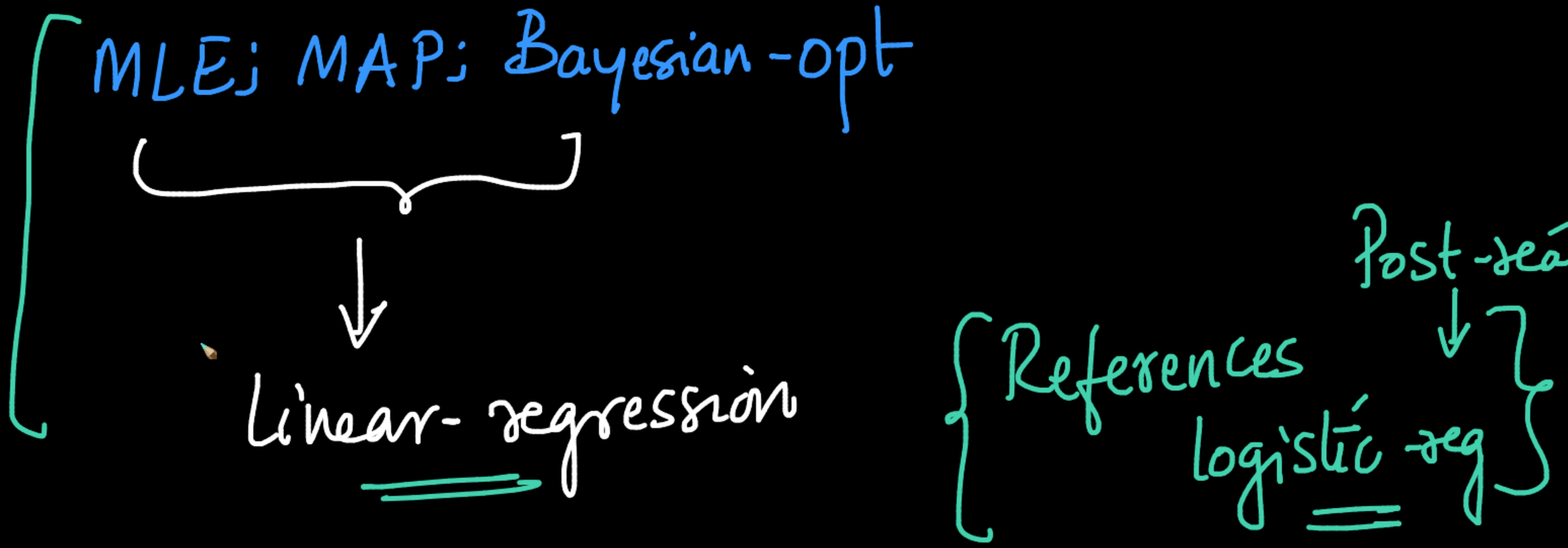
GD variants

→ Strong assumption about mixture of Gaussian distb underlying the data

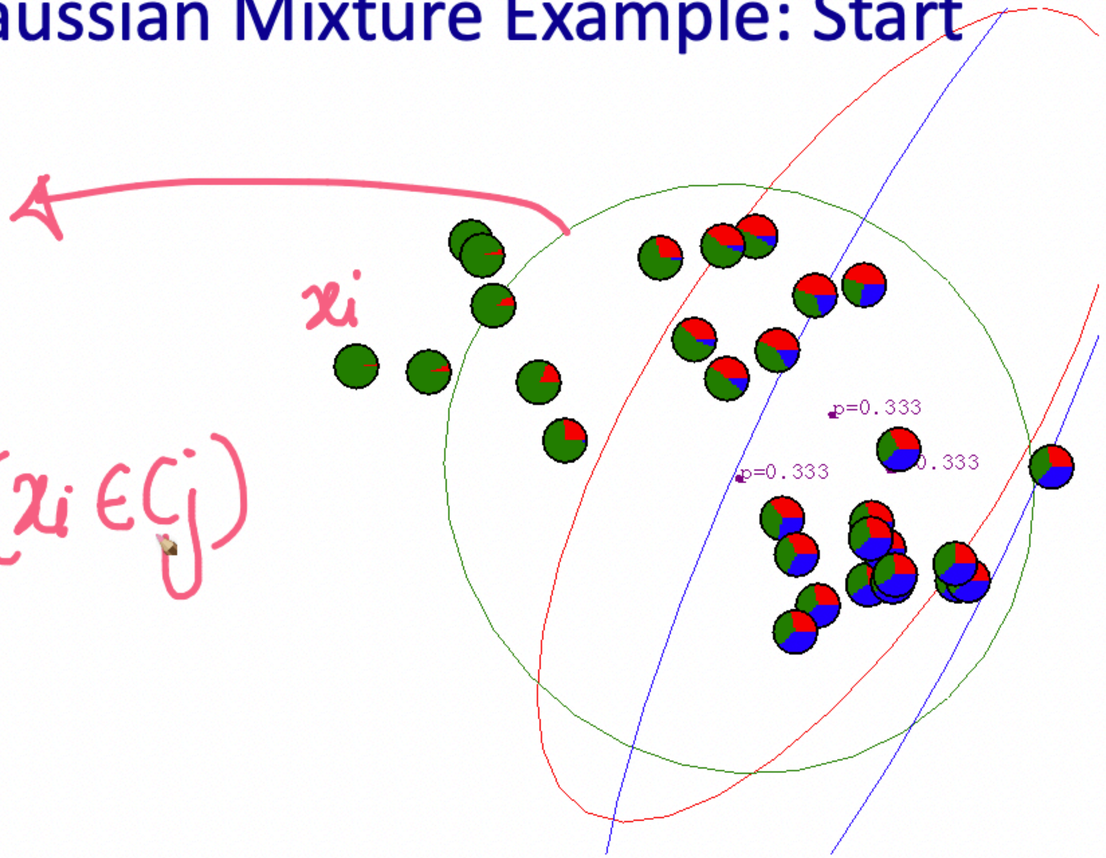
✓ [* → ✓ Generative model

(revisit) MLE : best θ that generated the D

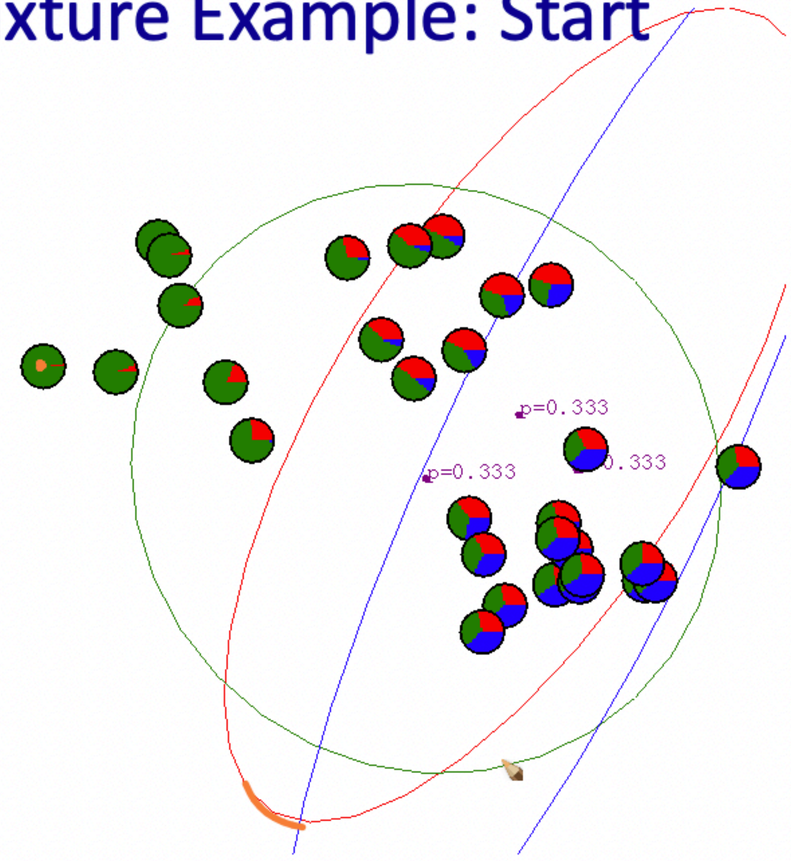
× EM - also...


MLE; MAP; Bayesian-opt
Linear-regression
Post-read
References
logistic-reg

Gaussian Mixture Example: Start



Gaussian Mixture Example: Start



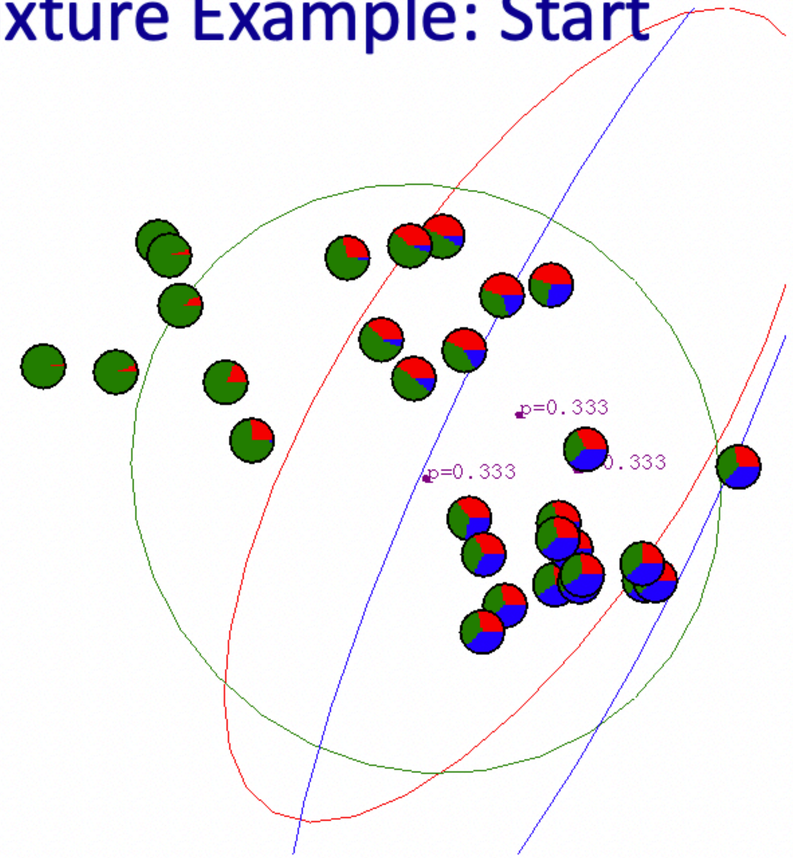
Exp: soft assign

Max: recompute μ^j, Σ^j

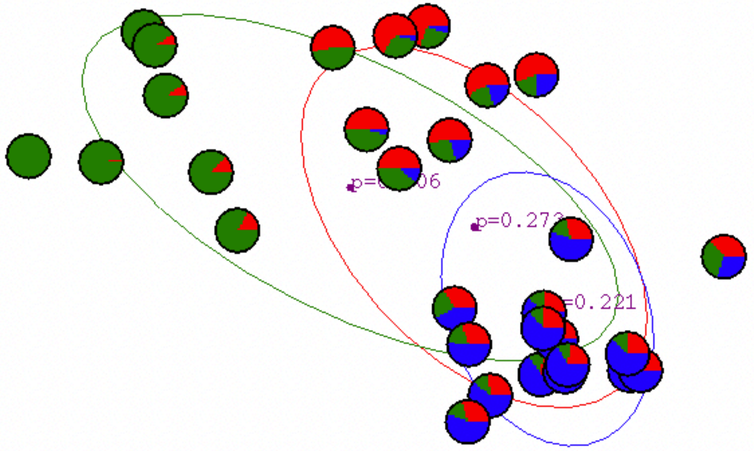
random μ_d^j, Σ_{dxd}^j

$j: 1 \to 3$

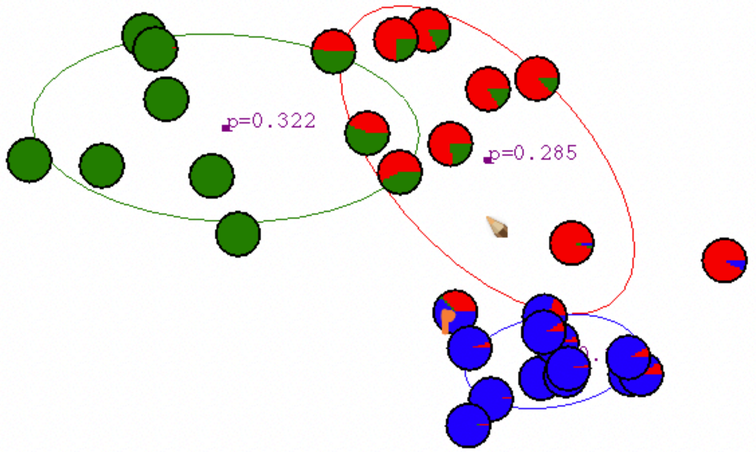
Gaussian Mixture Example: Start



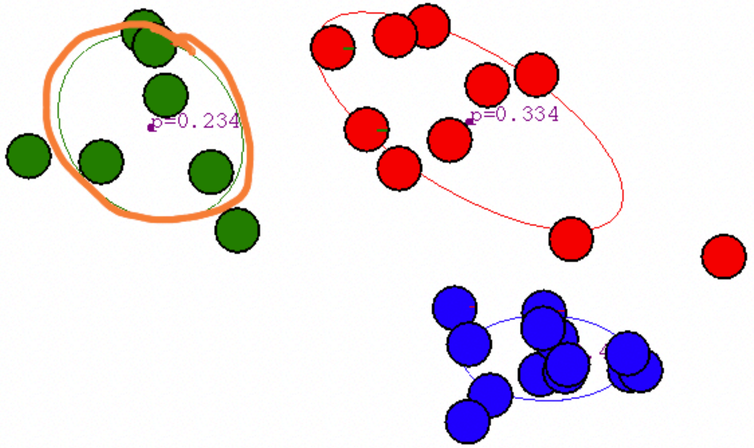
After first iteration



After 5th iteration



After 20th iteration



Suggestion

— MLE / optm2n → revisit

— EM

} revise

Misc. Topics

L_2-reg :

- Ridge-regression $= L_2\text{-reg} + \text{Linear reg}$
- Tikhonov reg ...

$$\min_{w_j} \; \text{loss} + \lambda \; L_2\text{-reg}$$

loss: log-loss, 0-1 loss, hinge-loss, Huber-loss

$$L_2\text{-reg} = \sum_{j=1}^{d} w_j^2$$

L_2-norm

weights of useless features to become small

Why L_2-reg in Linear-regression ?

↓

reduce overfitting

L_1-reg or Lasso

L_1-norm of w

sq. loss

$$\min_{w_j} \text{loss} + \lambda \sum_{j=1}^{d} |w_j|$$

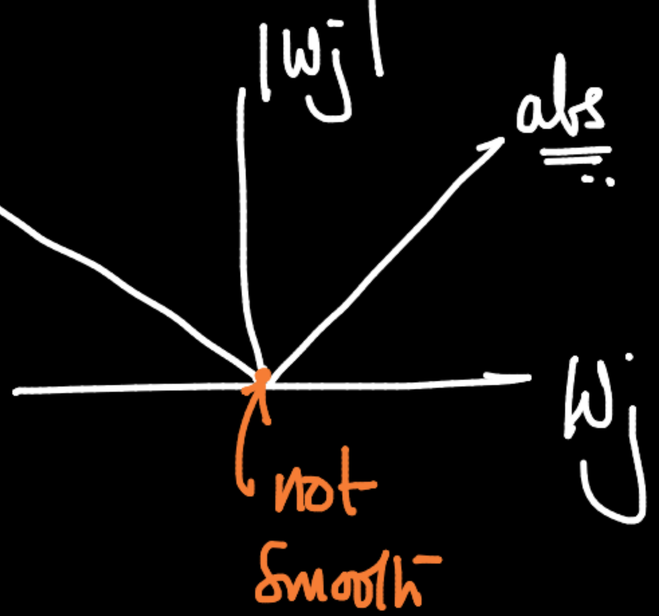
Is there a problem here?

$$L_1\text{-norm}(w) = \sum_{j=1}^{d} |w_j|$$

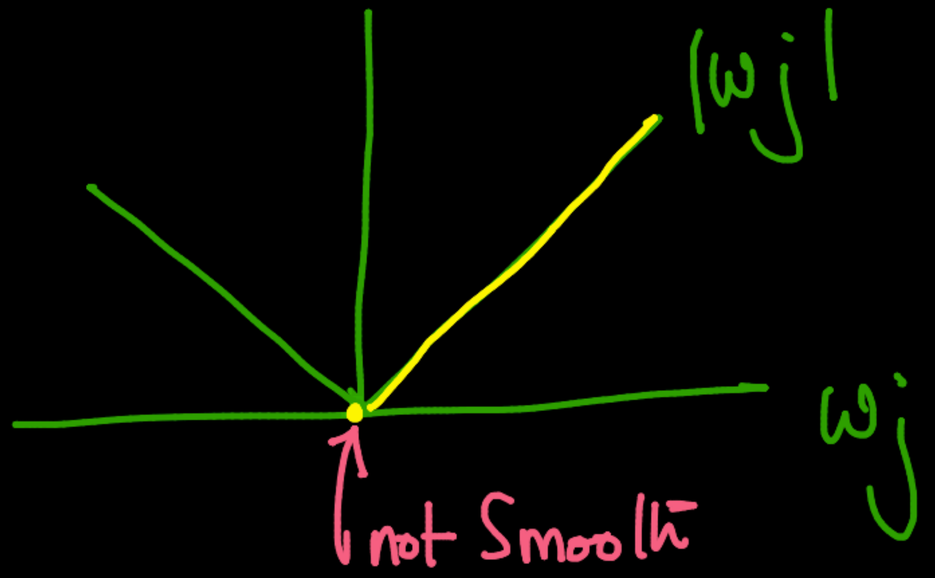
$$\min_{w_j} \underbrace{\sum_{i=1}^{n} \left(y_i - (w^T x_i + w_0)\right)^2 + \lambda \sum_{j=1}^{d} |w_j|}_{\mathcal{L}}$$

HINT: GD

$$\left\{ \frac{\partial \mathcal{L}}{\partial w_j} \;\middle|\; w_j = 0 \text{ is not defined} \right.$$



fix it:



der-abs(wj)

$$\frac{d|w_j|}{dw_j} = \begin{cases} +1 & \text{if } w_j > 0 \\ -1 & \text{if } w_j < 0 \\ 0 & \text{if } w_j = 0 \end{cases}$$

hack

$$\frac{d\,|w_j|}{dw_j} \;@\; w_j = 0$$

$= 0$ (why is it OK?)

↳ $\because w_j = 0$ means that f_j is useless

⇒ we can keep $w_j @ 0$

GD!

→ init all w_j's randomly + non-zero

→ update w_j's if some $w_j = 0 \Rightarrow f_j$ is useless

$$w_j^{new} = w_j^{0} - \eta \frac{\partial L}{\partial w_j}\Big|_{w_j^{0}}$$

$w_j^{0} = 0$ ✓

$\frac{\partial |w_j|}{\partial w_j}$ at 0 ← (OK) is not defined

L_1-reg: → results in a Sparse soln ⇒ W is sparse

[all less useful features weights become 0

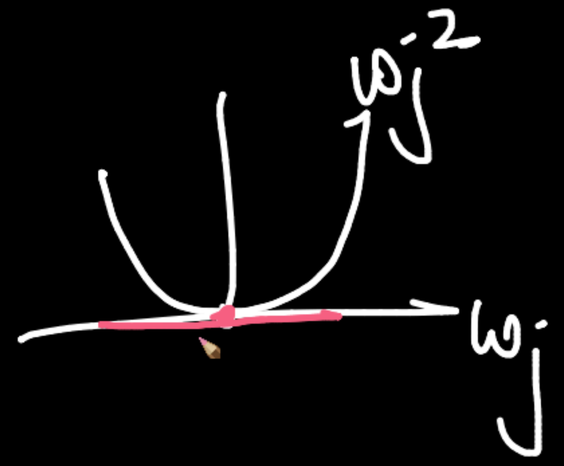
Why?

L_2-reg: → all less useful features weights become very small & close to 0

but $\neq$ ten not exactly equal to 0

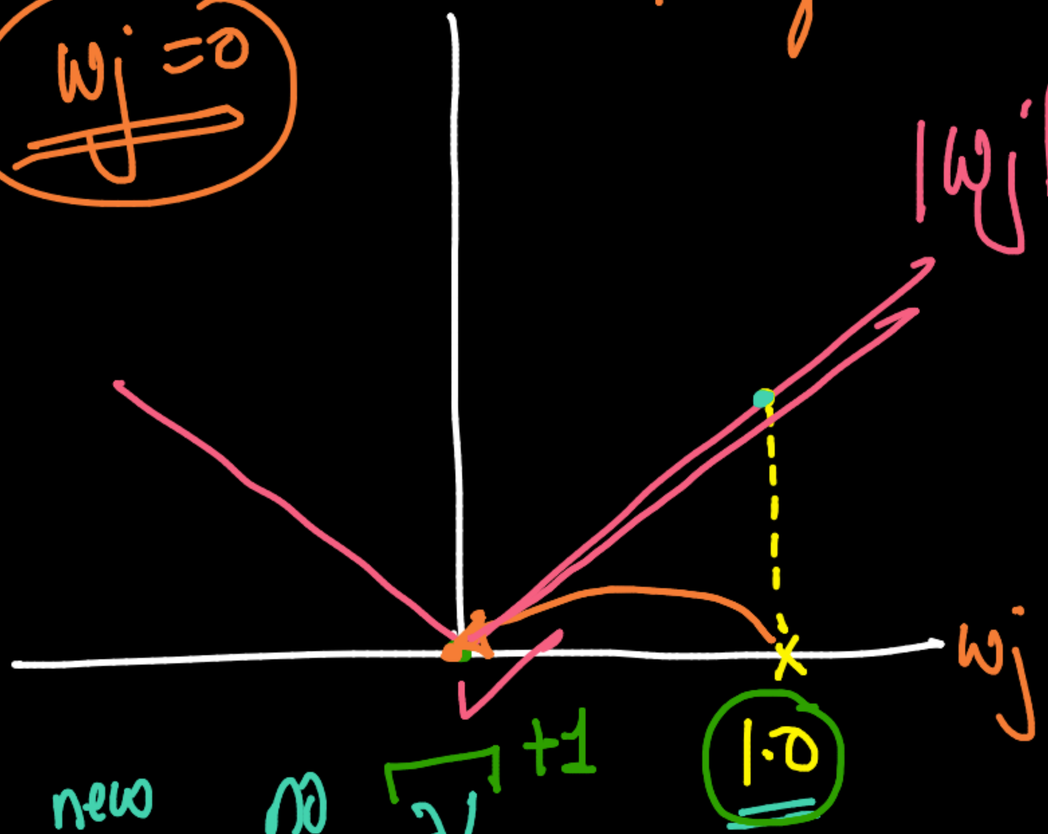
$$\left.\frac{d|w_j|}{dw_j}\right|_{w_j=0} = 0$$

$$\left.\frac{d\, w_j^2}{dw_j}\right|_{w_j=0} = 0$$



L1-reg

$w_j = 0$



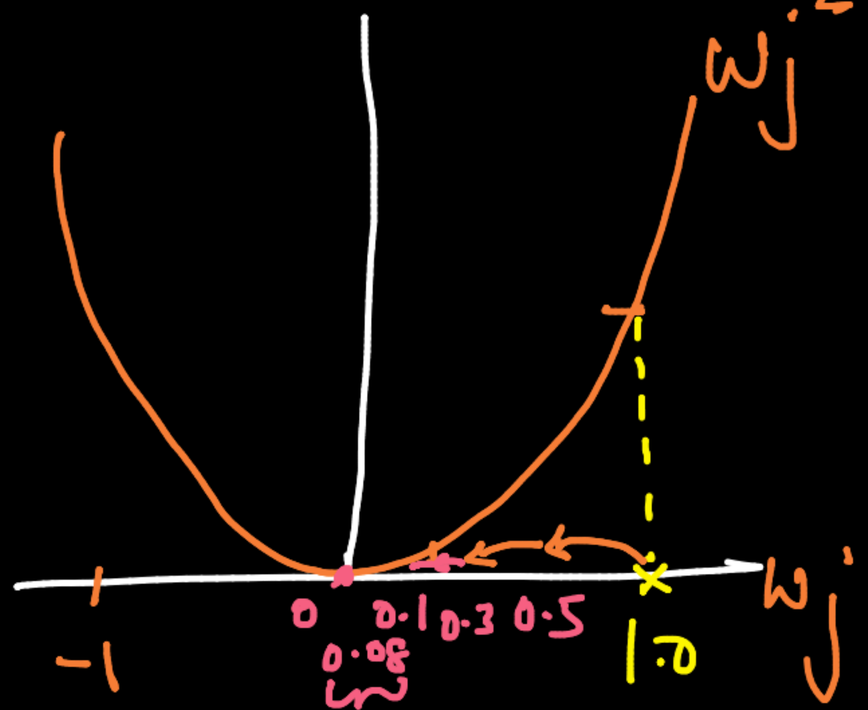
$$w_j^{new} = w_j^{old} - \eta \frac{\partial L}{\partial w_j}$$

$\frac{\partial L}{\partial w_j} \rightarrow 0.5$ (let)

Let's assume f_j is a useless feature

geometric

L2-reg



$$\frac{\partial L}{\partial w_j} = 2w_j$$

In L_2-reg as we approach '0'
the gradient also reduces

$$w_j^{new} = w_j^{old} - \eta \frac{\partial L}{\partial w_j}$$

L_1-reg → better Sparsity than L_2-reg

disadv: (1) dim #datapoints

$d > n$ (genomics)

↳ L_1-reg will have utmost n non-zero features even if $> n$ features are useful

(2) highly Correlated features

Eg → one feat non-zero
all other feat : zero

↓

using weights fw
F-1
(mess-up)

After 5th iteration