we want find w_i that minimize ~~error~~ loss function (squared error)

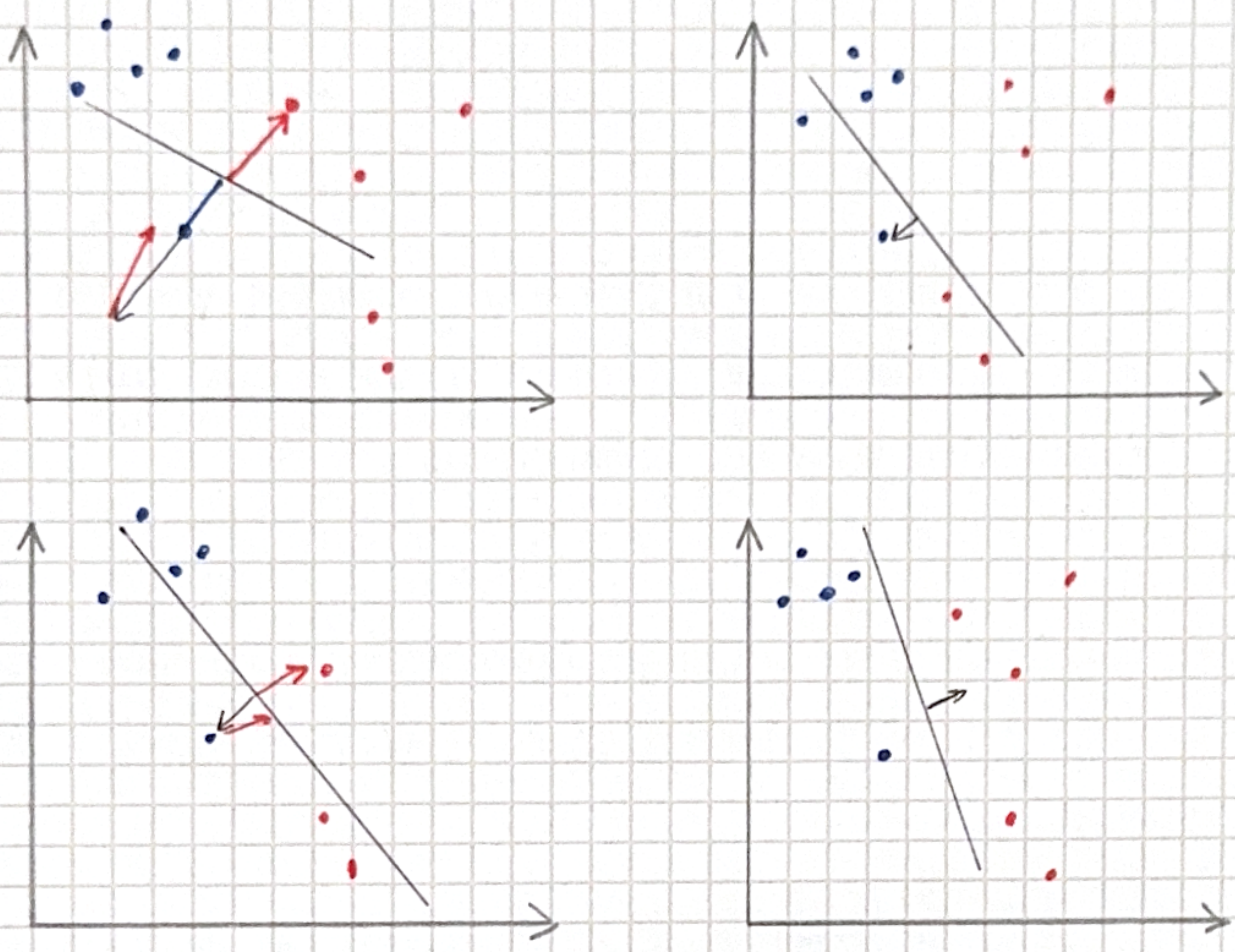
$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - o_n)^2 = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T x_n)^2$$

$$\frac{\partial E_D(w)}{\partial w} = \sum_{n=1}^{N}(t_n - w^T x_n)(-x_{i,n})$$

we find w^* in a sequential way by appling

$$w_i \leftarrow w_i + \Delta w_i \quad \text{whe} \quad \Delta w_i = -\eta\frac{\partial E}{\partial w_i} = \eta\sum_{n=1}^{N}(t_n - w^T x_n)x_{i,n}$$

2.



ML EXAMS 12/02/2018 A

EX. 1

1. In a Reinforcement Learning problem we have a dataset $D = \{\langle x_0, a_1, r_1, x_1, \ldots, a_n, r_n, x_n \rangle\}_{n=1}^{N}$ and we want learning behavior $\pi: S \to A$, meaning we want find the optimal policy function that maximize the reward
in RL problemes we don't have output and input in ~~this~~ tuple action, reward, state

2. for each x, a initialize table entry $\hat{Q}_{(0)}(x, a) \leftarrow 0$
observe current state x
for each time $t = 1, \ldots, T$ (until termination condition)
   - chose an action a
   - execute action a
   - observe the new state x'

- collect the immediate reward r
- update the table entry for $\hat{Q}(x,a)$

$$\hat{Q}(x,a) \leftarrow r + \gamma \max (\hat{Q}_{(t-1)}(x',a'))$$

- $x \leftarrow x'$

optimal policy $\pi^*(x) = \underset{a \in A}{\operatorname{argmax}} (\hat{Q}_{(t)}(x,a))$

EX. 2

DROPOUT: randomly ignore network units with some probability p at each iteration

EARLY-STOPPING: we stop the train for avoid overfitting. we can stop for example when validation loss doesn't decrease

EX. 3

1. SVM for classification aims to maximum margin for providing better accuracy

given a dataset $D = \{(x_n, t_n)\}_{n=1}^{N}$ and x_k the closest point on the hyperplane $h = w^T x + w_0 = 0$

the margin is $\frac{|y(x_k)|}{||w||}$ and is computed as:

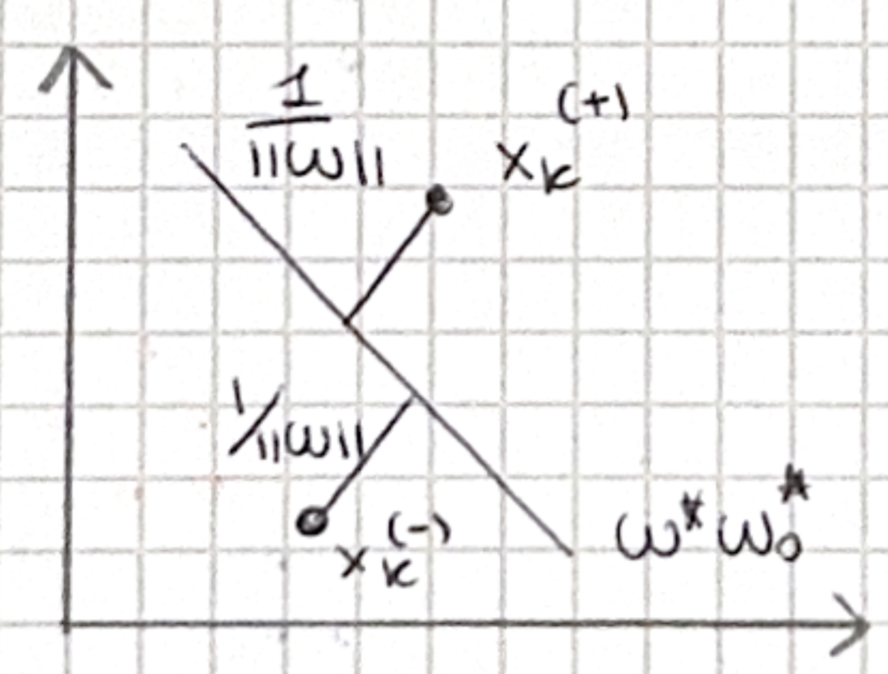
$$\min_{n=1,\dots,N} \frac{|y(x_n)|}{||w||} = \frac{1}{||w||} \underset{n=1,\dots,N}{\operatorname{argmin}} [t_n (w^T x_n + w_0)]$$

using the property $|y(x_n)| = t_n\, y(x_n)$

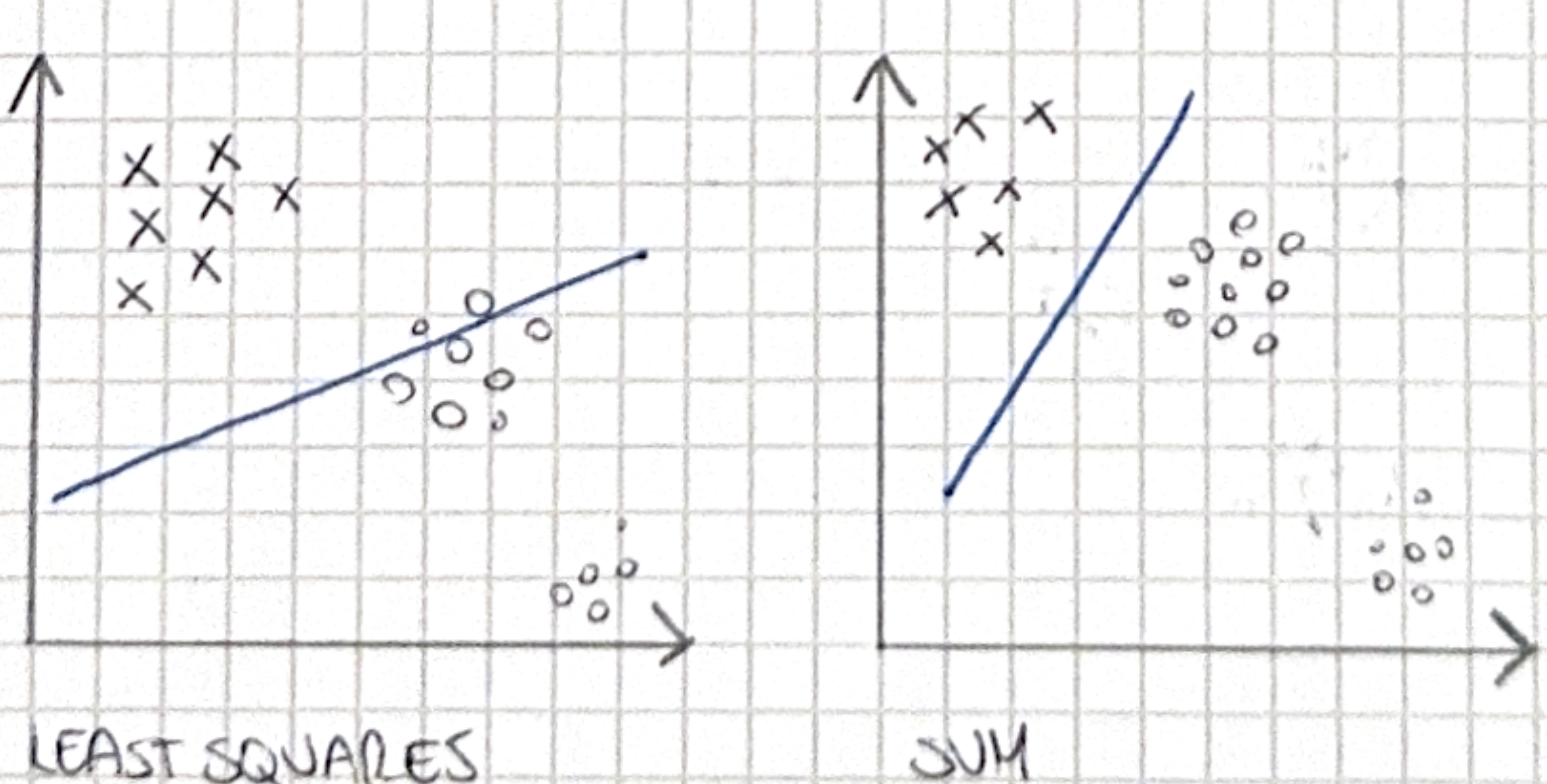
so the maximum margin is computed as:

$$w^*, w_0^* = \underset{w, w_0}{\operatorname{argmax}} \frac{1}{||w||} \min_{n=1,\dots,N} \left(t_n (w^T x_n + w_0) \right)$$

geometric representation:



2.



3. we prefer SVM because if data is linear separable ~~we~~ the result will be a perfect line that separate the sample ~~on~~ with other models ~~...~~ we are not sure about for example in least squares can happen ~~too~~ that with outliers the the models can misclassifies the elements

EX. 4

1. confusion matrix report how many times an instance of class c_i is classified in class c_j

2.

	low	Med	High
low	10	40	50
med	20	20	80
high	30	5	65

↓ TRUE VALUE \ PREDICTED →	low	Med	high
low	10%	40%	50%
Med	20%	20%	80%
hig	30%	5%	65%

3. the accuracy is the sum of element in the diagonal divided by the num of all elements

95/300 = 31%

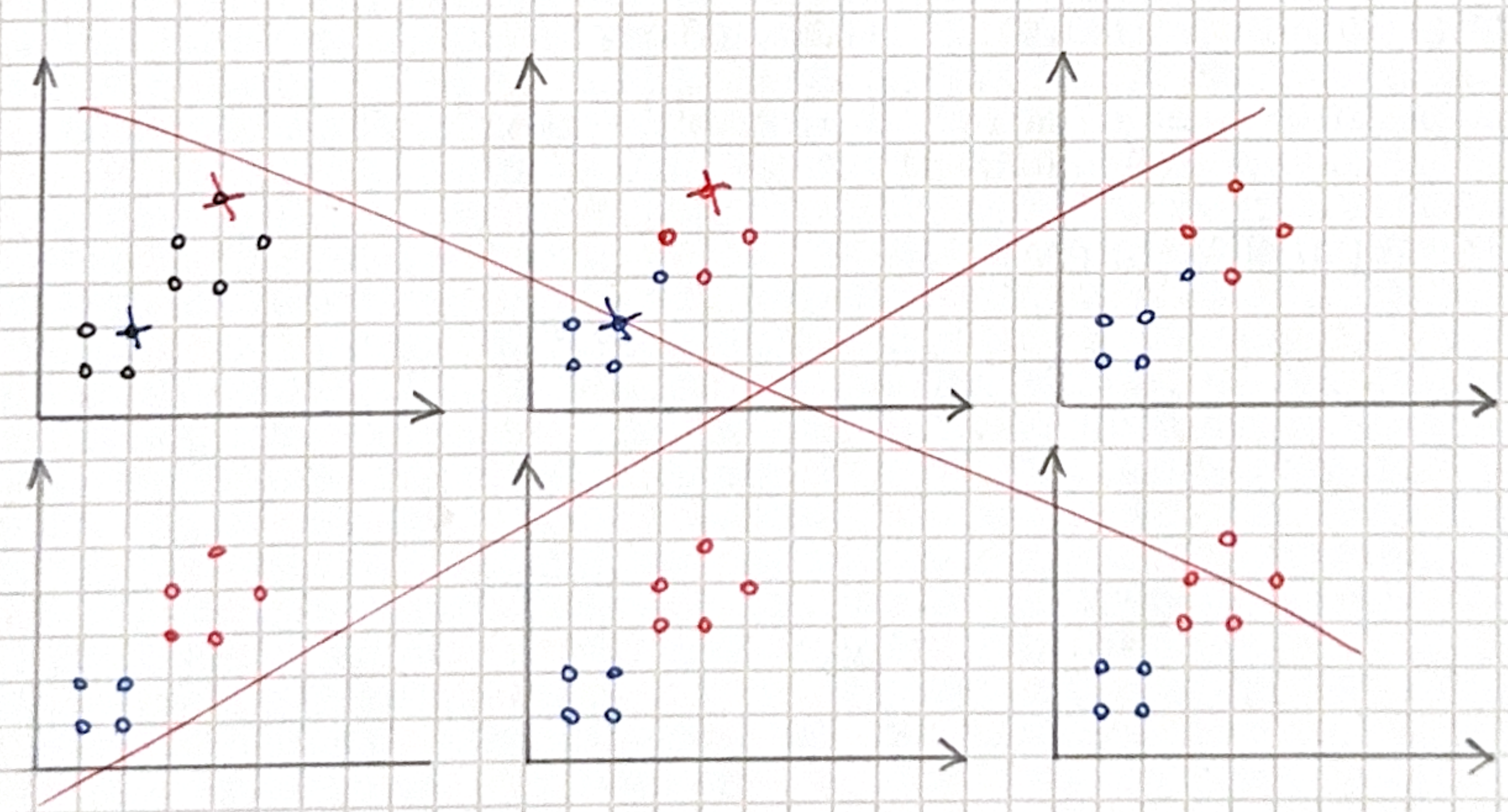
EX. 5

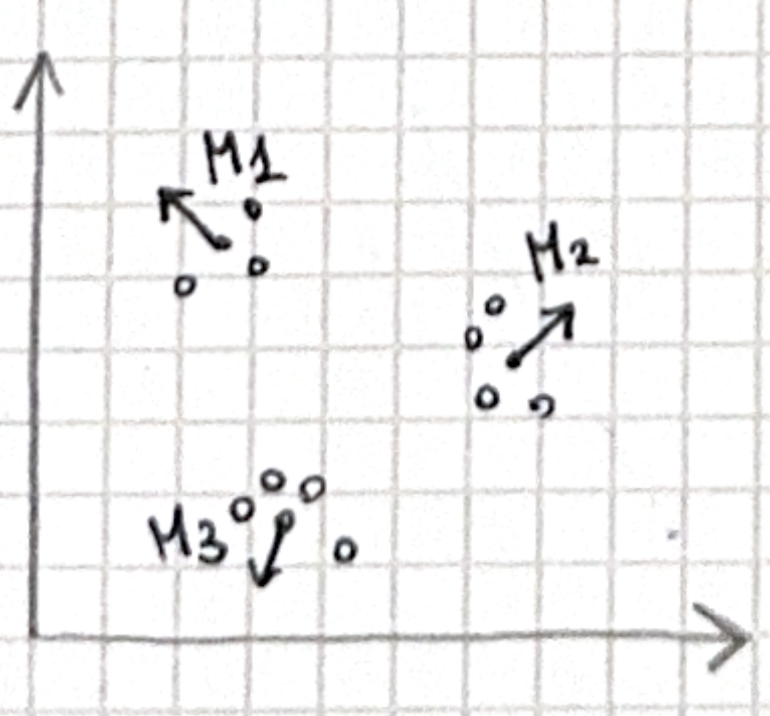
1. mixed probability distribution P formed by k different Gaussian distribution

$$P(X) = \sum_{k=1}^{K} \pi_k N(x, \mu_k, \Sigma_k)$$

where π_k: prior probability μ_k: mean Σ_k: covariance matrix

2.





K=3

3. the unknown prior probabilities = 2 . 3 known mean vector of size 2 · 3 unknown covariance matrix with 3 unknown parameters each one size $2+(3\cdot2)+(3\cdot3) = 17$

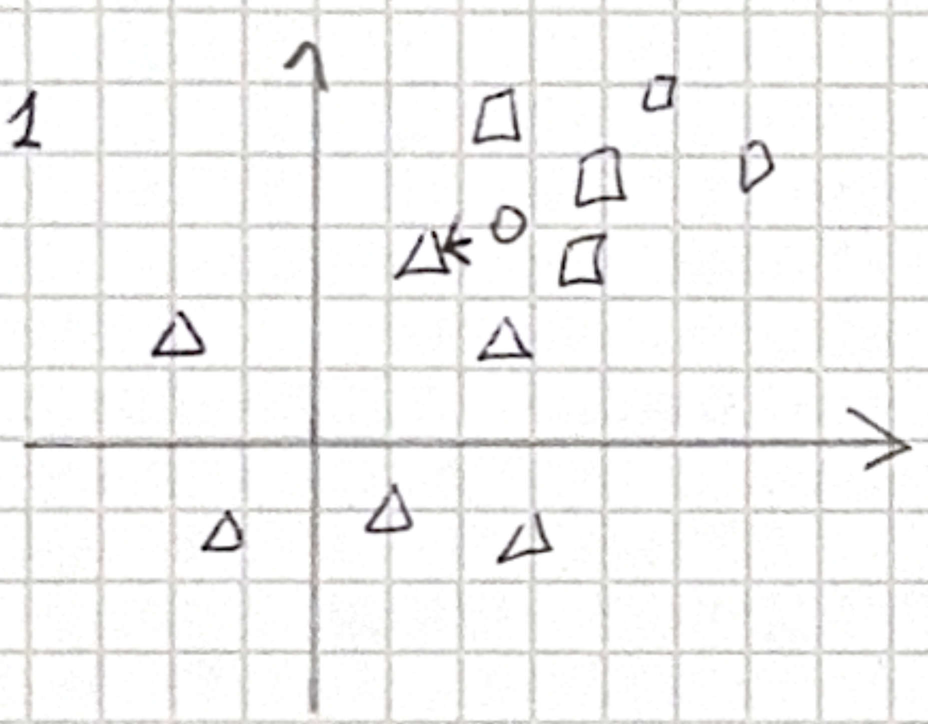
EX. 6

1. classification problem $f: X \to C$ with dataset $D = \{(x_n, t_n)\}_{n=1}^{N}$
classification with KNN
① find k-nearest neighbors of new instance x
② Assign to x the most common label among the majority of neighbors
likelihood of class c of new instance x is

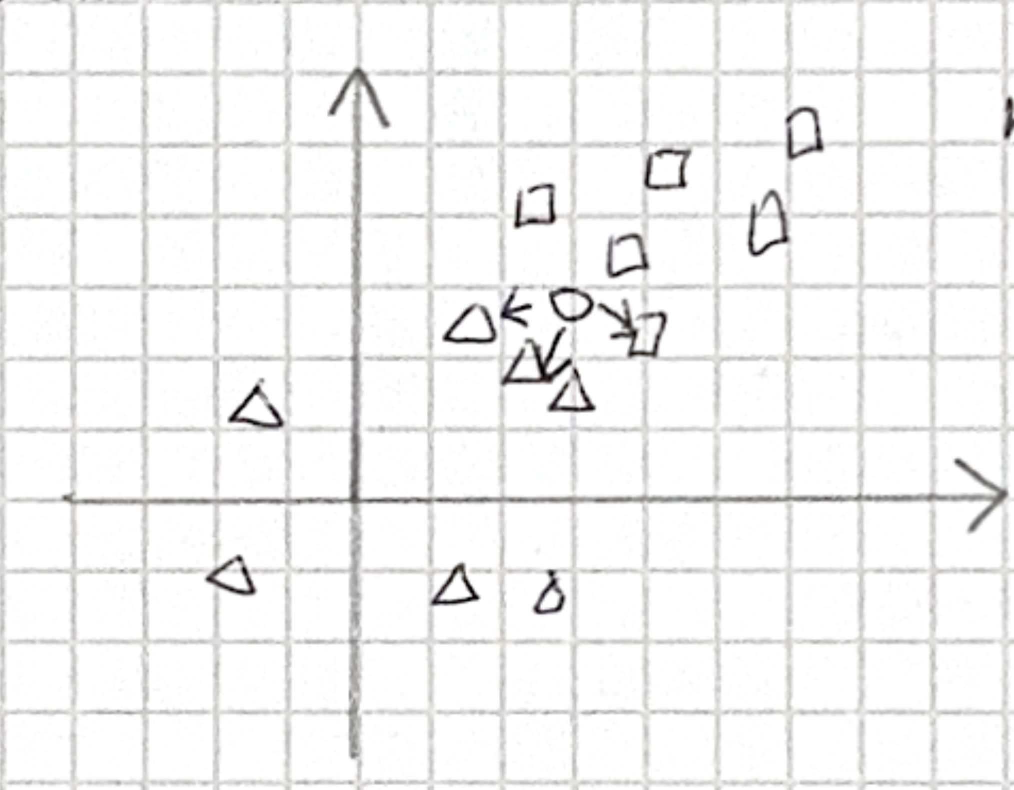
$$P(c \mid x, D, k) = \frac{1}{k} \sum_{x_n \in N_k(x_n, D)} \mathbb{I}(t_n = c)$$

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e = \text{true} \\ 0 & \text{otherwise} \end{cases}$$

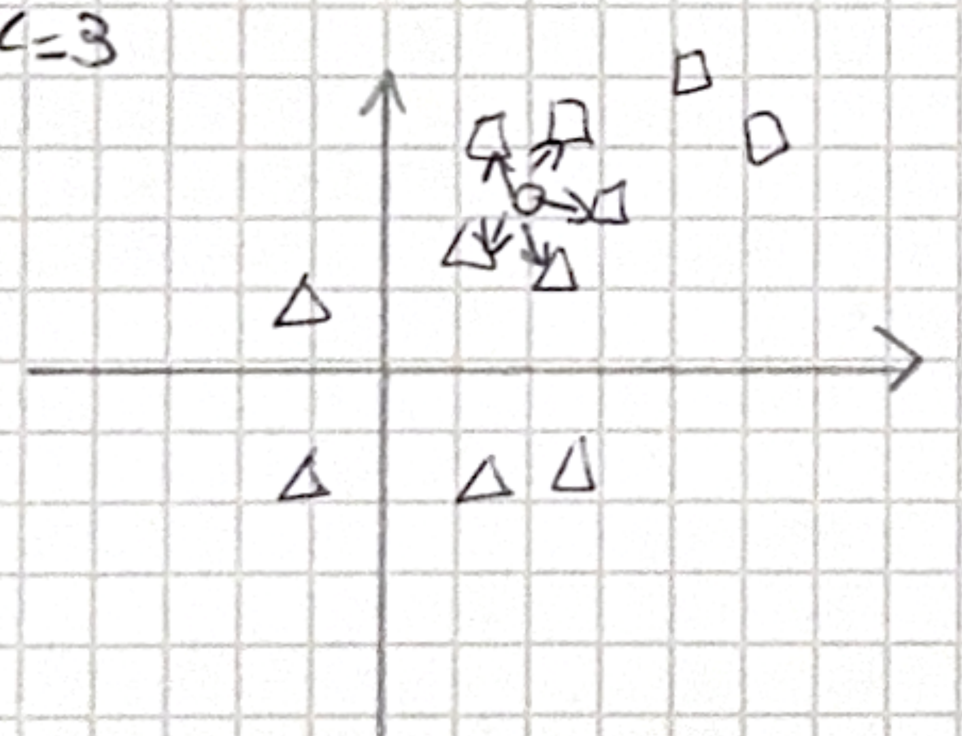
2. k = 1



k=3



K=3



· for k = 1
$P(c \mid x, D, k) = \frac{1}{1} \cdot 1 \, (\Delta) = 1 \to \Delta$

· for K = 3
$\square = 1$
$\Delta = 2$
$P(\square \mid x, D, k) = \frac{1}{3}(1) = 0.33$
$P(\Delta \mid x, D, k) = \frac{1}{3}(2) = 0.66$

· for k = 5
$\square = 3$ $\quad P(\square \mid x, D, k) = \frac{1}{5} \cdot (3) = 0.6$
$\Delta = 2$
$P(\Delta \mid x, D, k) = \frac{1}{5}(2) = 0.4$