3. Boc is an optimal classifier if the hypothesis space is not too large and it has an analytical solutions otherwise is not practical and bcuse Naive and Naive Bayesian classifier

EX. 6

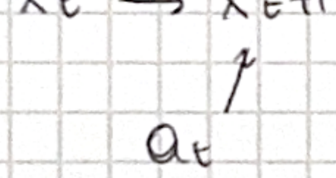
once the current state is known the evolution of a dynamic system doesn't depend on previous state actions and observations. the current state contains all information needed to predict the future

future state are conditionally independent of past states and past observations statistically independent

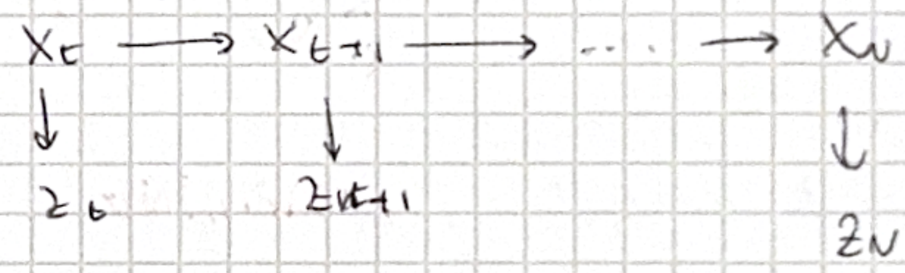
MDP can be describe $\langle X, A, f, r \rangle$ X: finite set of states
A: finite set of actions $f: X \times A \to X$ transition function and
r: reward function

HMM can be describe $\langle X, Z, \Pi_0 \rangle$ where X, Z is set of observations
and Π_0 initial distribution
MDP not the property of fully observability

MDP



HMM



ML EXAMS 15/10/2019

EX. 1

1. how many times an element of class c_i in classifies are in the class ej

2.

\	A	B	C
A	40	50	10
B	10	80	10
C	5	5	90

\	A	B	C	← predicted
A	40%	50%	10%	
B	10%	80%	10%	
C	5%	5%	80%	

↳ true values

3. the accuracy in a confusion matrix is the num of elements in the diagonal divide by the all of elements of matrix

accuracy = (40 + 80 + 80)/300 = 0.7

EX 2

1.

for solve this task I will use linear regression

we know that $y(x,w) = \sum_{j=1}^{N} w_j \phi_j(x)$

that is a linear model in parameter w. we know that $t = y(x,w) + \varepsilon$

with ε solai hive noise (Gaussian)

$\Rightarrow p(\varepsilon|\beta) = N(\varepsilon|0,\beta^{-1}) \Rightarrow p(y|x_i, x_n, w, \beta) = N(y|y(x,w), \beta^{-1})$

if we use iid hypothesis $p(\{y_1,\dots,y_n\}|x_1,\dots,x_n,w,\beta) =$

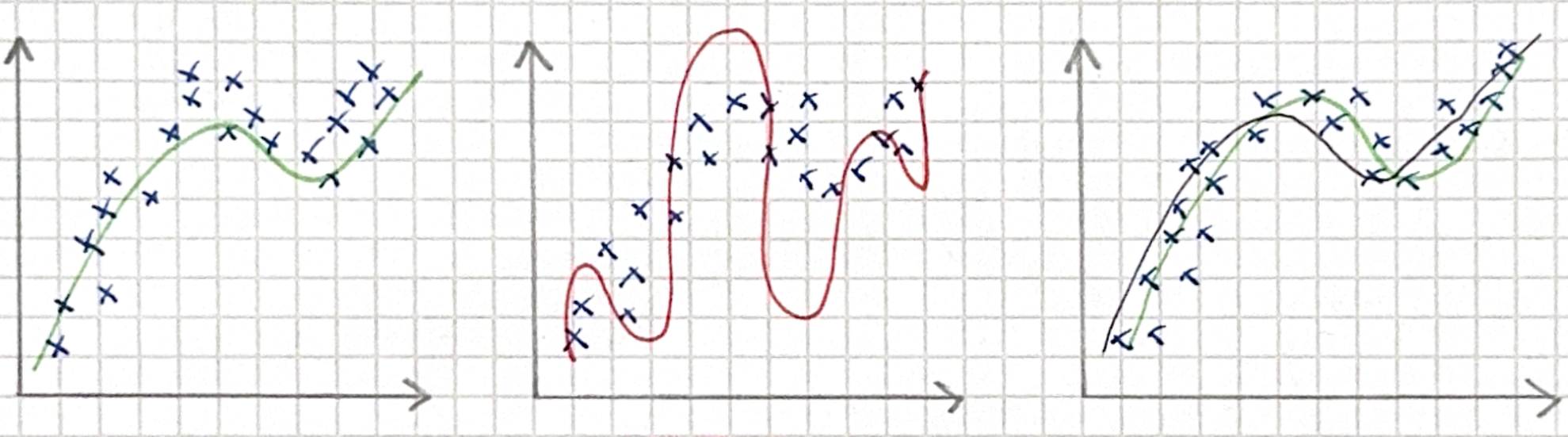
$= \prod_{n=1}^{N} N(y_n|w^T\phi(x), \beta^{-1}) = -\frac{\beta}{2}\sum_{n=1}^{N}(y_n - w^T\phi(x_n))^2 - \frac{N}{2}\ln(2\pi\beta^{-1})$

the error function is $E_D(w) = \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T\phi(x_n))$ and we

want fing $\arg\min E_D(w)$

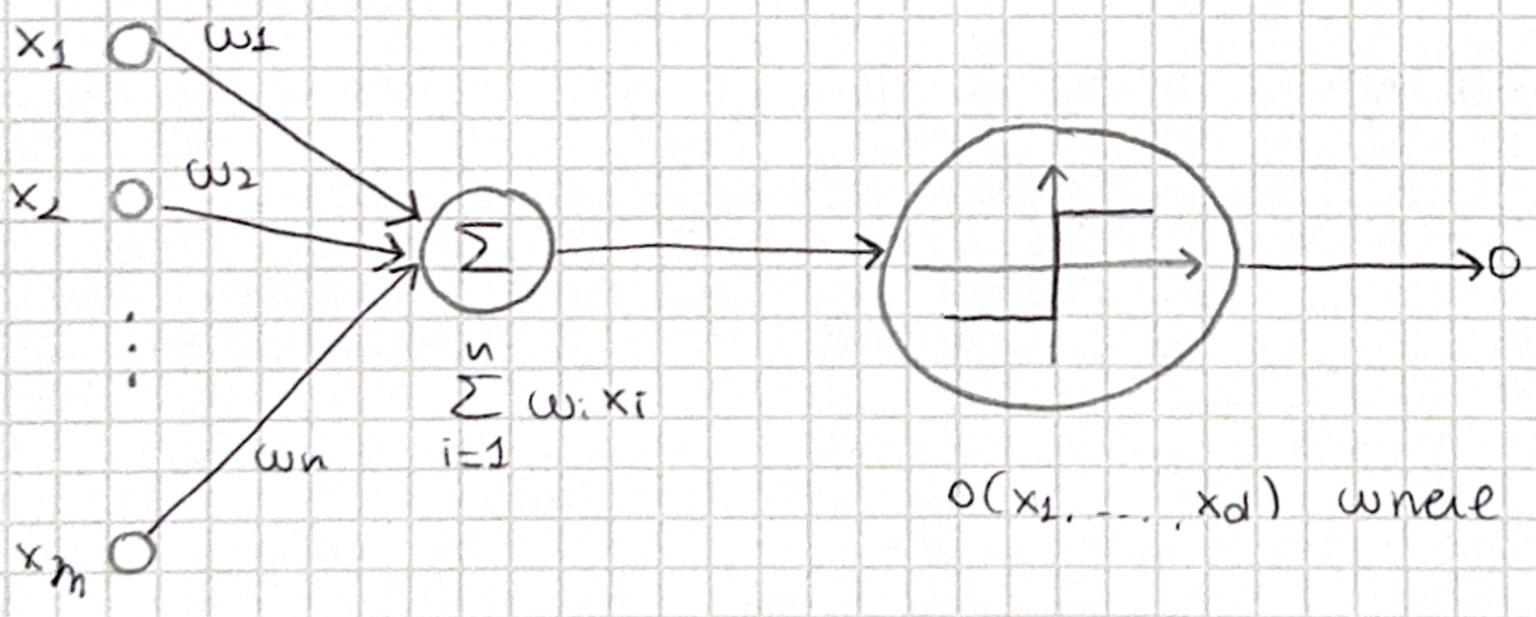
2 we can apply regularization $w^* = \arg\min E_D(w) + \lambda E_w(w)$

with $\lambda > 0$ and $E_w(w) = \frac{1}{2} w^T w$



EX. 3

1.



$o(x_1,\dots,x_d)$ where $o(x_1,\dots x_d) = \begin{cases} 1 \text{ if } \sum_{i=1}^{N} w_i x_i > 0 \\ 0 \text{ otherwise} \end{cases}$

we want minimite the function:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - o(n))^2 = \frac{1}{2}\sum_{n=1}^{n}(t_n^n - xw^T x_n)^2$$

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

2) we want to find w^* sequentially:

$\hat{w} \leftarrow \hat{w} + \eta \Delta w_i$ where $\Delta w_i = -\eta \sum_{n=1}^{N} (t_n - \text{sign}(w^T x_n))(x_{i,n})$

Δw_i can be update with

batch mini batch or sequential mode

3) if we use an large η is possible that will diverge otherwise

if η is too small the algorithm will never converge

but the algorithm is robust to local minima

EX. 4

1. SVM aims at maximum margin providing for better accuracy

the margin is $|y(x_n)| / \|w\|$ (smallest distance between x_k and h)

given a dataset D and hyperplane h the margin is computed

as

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

so the maximum margin is computed as:

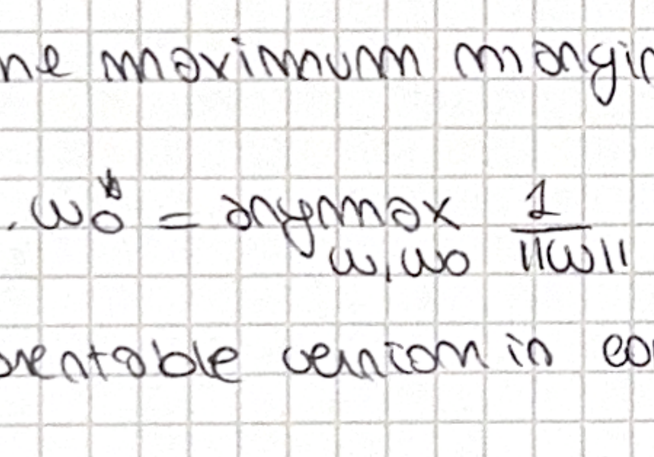
$$w^*, w_0^* = \arg\max_{w, w_0} \frac{1}{\|w\|} \min [t_n(w^T x_n + w_0)]$$

a more stable version is computed by averaging over all support vectors

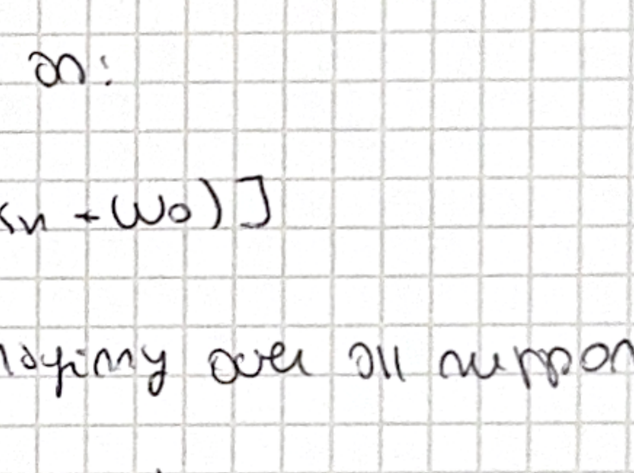
$$w_0^* = \frac{1}{|SV|} \sum_{x_k \in SV} \left(t_k - \sum_{x_j \in S} \hat{a_j} t_j x_k^T x_j \right)$$

where a_j are lagrangian optimizers results of primitation problem

2



least squares



SVM

also robust to outliers

EX. 5

1. given a dataset $D = \{(x_n, t_n)\}_{n=1}^{N}$ and a target function $f: X \to Y$

KNN is an instance based model:

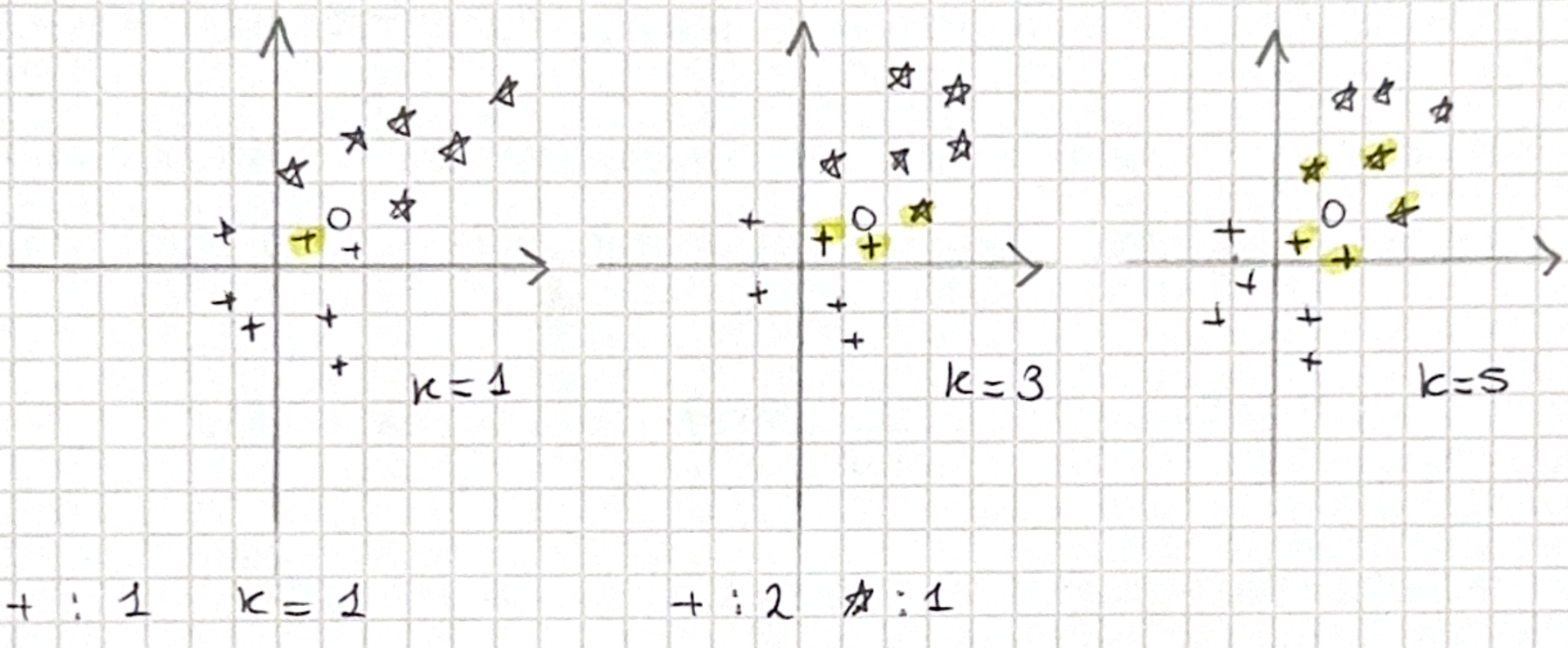
1. find k nearest neighbor of new instance x
2. assign to x the most common label among the majority of neighbors

likelihood of class c of new instance x is

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

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

2.



+ : 1 $\quad k = 1$ $\qquad$ + : 2 ☆ : 1

$P(+|x, D, 1) = \frac{1}{1}(1) = 1$ $\qquad$ $P(+|x, D, 3) = \frac{1}{3}(2)$ $\qquad$ $P(+|x, D, 5) = \frac{1}{5}(2) =$

classified as + $\qquad$ $P(☆|x, D, 3) = \frac{1}{3}(1)$ $\qquad$ $P(☆|x, D, 5) = \frac{1}{5}(3) =$

classified as + $\qquad$ classified as ☆

EX. 6

1. Boosting is an ensable method. the main idea is instead of train one ~~example~~ simple complex classifier we train different classifier and we combine them result in boosting we retraining models in repeated each models is based on the error of the previous model

2. ADABOOST

I. minimizing error function:

$$J_m = \sum_{n=1}^{N} w_n^{(m)} \mathbb{I}(y_m(x_n) \neq t_n)$$

$$\sum_{n=1}^{N} w_n^{(m)}$$