ML EXAMS 28/06/2024

EX. 1

1.1

if $C = C_1$ and $B = b_1$ ~~and~~ then YES ①

if $C = C_1$ and $B = b_2$ then NO ②

if $C = C_2$ and $A = a_1$ then YES ③

if $C = C_2$ and $A = a_2$ and $B = b_1$ then YES ④

if $C = C_2$ and $A = a_2$ and $B = b_2$ then NO ⑤

if $C = C_2$ and $A = a_3$ then NO ⑥

if $C = C_3$ then NO ⑦

1.2

an hypothesis h is consistent with a set of training examples D of target concept c if and only if $h(x) = c(x)$ for each training example $(x, c(x))$ in D

S_1 is consistent because of ③

S_2 is consistent because of ①

S_3 is NOT consistent because of ⑦

S_4 is consistent because of ②

S_5 is consistent because of ⑥

EX. 2

2.1

KNN is an instanced based model where given a classification problem $f : X \to C$ with dataset $D = \{(x_n, t_n)\}_{n=1}^{N}$

classification KNN

① find k nearest neighbors of new instance x

② assign to x the most common label among the majority of neighbor

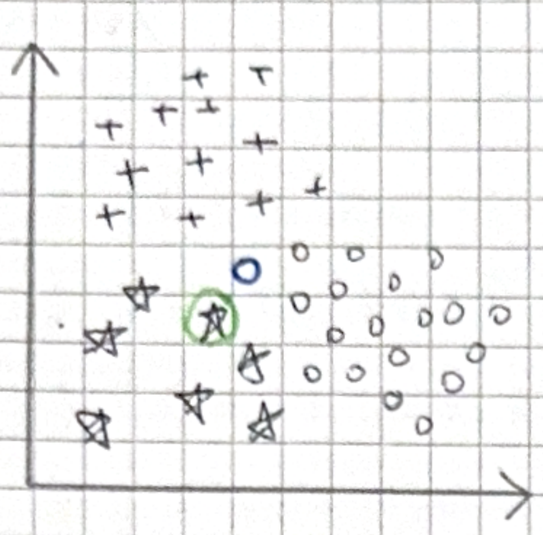
likelihood of class for new instance x

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

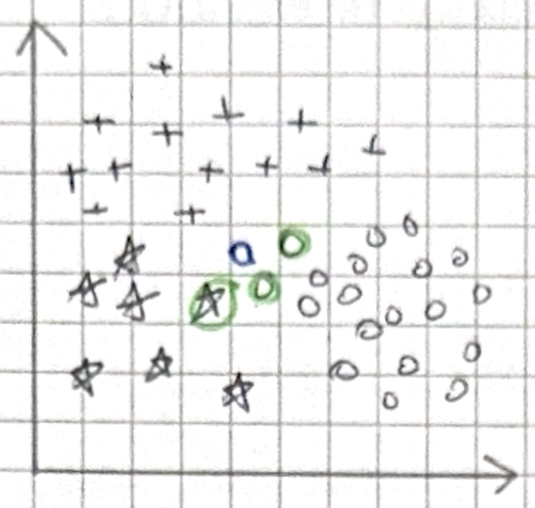
where $\mathbb{I}(e) = \begin{cases} 1 & \text{if } e = \text{true} \\ 0 & \text{otherwise} \end{cases}$

KNN, different respect other models doesn't require a training phase but require all dataset in order to decide which class new instance x belongs to

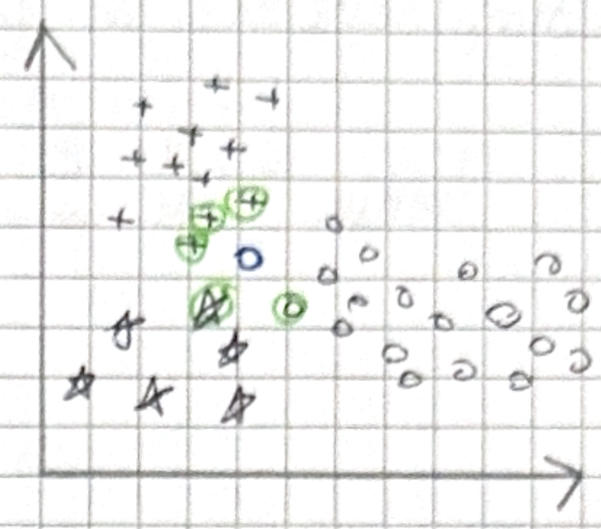
2.2



k=1 class = ☆



k=3 class = o



k=5 class = +

Ex. 3

3.1

both methods BAGGING and BOSTING are method based on concept of train multiple model the different between two is that in BAGGING we train model in a parallel way different in for BOOSTING where the model are train in sequencial way another different is that ~~bagging~~ Boosting use same dataset D for train m model with ~~Boost~~ ~~boostrap~~ bagging we generate m bootstrap dataset from D and we use each of them for train model.
final both methods use a voting scheme for find best model

3.2

BAGGING

given a dataset D

1. generate M bootstrap dataset from D
2. train each model with each bootstrap dataset
3. make prediction with voting scheme

$$y_{voting} = \frac{1}{M} \sum_{m=1}^{M} y_m(x)$$

* 3. $Y_m = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_n(x) \right)$

ADABOOST

1. initialize $w_n^{(1)} = 1/N$
2. for $m = 1, \ldots, M$

• train a weak learners $y_m(x)$ by minimizing the weighted error function

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

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

• evaluate $\varepsilon_m = \dfrac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$

• update weights

$$w_n^{m+1} \leftarrow w_n^m \exp[\alpha_m I(y_m(x_n) \neq t_n)]$$

$$\alpha_m = \ln\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)$$

*

for combine them we can use different $y_m = sign\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right)$
and apply voting scheme as for Bagging method

$$y_{VOTING} = \frac{1}{M} \sum_{m=1}^{M} y_m$$

EX. 4

4.1

SVM aims to maximum margin for better accuracy a margin is defined as $|y(x_n)| / \|w\|$ so the goal of SVM is:

$$\arg\max \left(\min \left(\frac{|y(x_n)|}{\|w\|} \right) \right) = \arg\max \frac{1}{\|w\|} \min \left[t_n (w^T x_n + w_0) \right]$$

using property $|y(x_n)| = t_n y_n(x)$

this is an optimization problem that can be solved using lagrangian multiplier a^*.

for a better estimation we will be useful averaging over all SV

$SV = \{ x_k \in D \mid t_k y(x_k) = 1 \}$ in this way we have

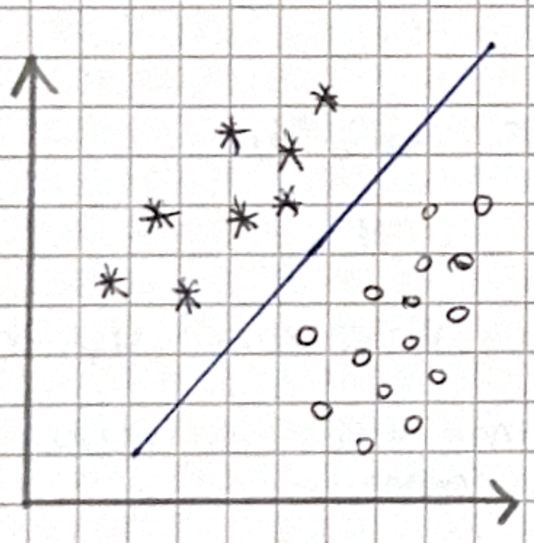
$$w_0^* = \frac{1}{|SV|} \sum_{x_k \in SV} \left(t_k - \sum_{x_j \in S} \alpha_j^* t_j x_k^T x_j \right)$$

$$y(x, w) = sign\left(\sum_{x_k \in SV} a_k^* t_k x'^T x_k + w_0^* \right)$$

4.3

is better use SVM when we have separable dataset because using the margin for find best solution make this methods robust the outliers so we will have better performances

4.2



EX. 5

5.1

$f: \mathbb{R}^5 \to \{C_1, C_2, C_3\}$

$$P(C_1|X) = \frac{P(X|C_1)P(C_1)}{P(X)} = \frac{P(X|C_1)P(C_1)}{P(X|C_1)P(C_1) + P(X|C_2)P(C_2) + P(X|C_3)P(C_3)}$$

$$P(C_2|X) = \frac{P(X|C_2)P(C_2)}{P(X)} = \frac{P(X|C_2)P(C_2)}{P(X|C_1)P(C_2) + P(X|C_2)P(C_2) + P(X|C_3)P(C_3)}$$

Ex. 5

5.1

MDP = $< X, A, \delta, r >$

X = rooms δ: $X \times A \to X$ is transition function

A = clean and move r: reward function

5.2

considering $N = 3$ we can derive the optimal policy

N=1	N=2	N=3

π → policy have to maximize the cumulative reward in particular

if $r(s,a) = +1$ clean room

$r(s,a) = -2$ move to room

$r(s,a) = 0$ otherwise

$\pi = \{((N=2, clean), move\ to\ N_2), ((N1, dirty), clean), ((N2, dirty), clean)$
$((N=2, clean), move\ to\ N3), ((N=3, dirty), clean), ((N=3, clean), stop)\}$

initial state = N=1

Ex. 6

6.1

$$P(x) = \sum_{n=1}^{3} \pi_n N(x, \mu_N, \Sigma_N)$$

x: input data μ_N = mean Σ: covariance matrix

weight of component π_j $k=3$ but $\sum_k \pi_k = 1 \to k-1=2$

μ_n is a vector in two dimension so # params of $\mu_n = 6$

Σ_k is a symmetric covariance matrix $d=2$ so # params $d \times \frac{(d+1)}{2} = 3$