IB031 – příklady k procvičení

POZOR! Zkouškové příklady nemusí být přesně stejné!

1. Prezentujte (formálně) kompletní inferenční algoritmus K Nearest Neighbors (KNN).

   Není nutné specifikovat, jak přesně určujete nejbližší sousedy.

2. Uvažujte dataset $D = \{((0,0),1),((1,0),0),((0,1),0),((1,1),1\}$. Uvažte KNN algoritmus s konstantou K (počet uvažovaných sousedů) a Euklidovskou vzdáleností k sousedům. Pro které body ve čtverci s rohy $(0,0),(1,0),(0,1),(1,1)$ bude KNN vracet třídu 1 pokud

   - $K = 1$
   - $K = 2$
   - $K = 3$

   (V případě, že nelze rozhodnout mezi třídami 1 a 0, preferujte třídu 1.)

3. Prezentujte (formálně) kompletní algoritmus ID3, ve kterém bude rozhodnutí o nejlépe klasifikujícím atributu dáno pomocí impurity decrease (*ImpDec*).

4. Demonstrujte kompletní výpočet algoritmu ID3 pro trénink rozhodovacích stromů (decision trees) s impurity decrease (*ImpDec*) na následujícím datasetu:

index	X	Y	class
1	1	1	Yes
2	1	1	Yes
3	1	1	Yes
4	1	0	Yes
5	0	0	Yes
6	1	0	No
7	0	1	No
8	0	1	No
9	1	0	No
10	0	0	No

   Dataset obsahuje dva atributy X, Y, oba s hodnotami v $\{0,1\}$, a třídu (class) s hodnotami $\{\text{Yes}, \text{No}\}$.

5. V rámci předchozího příkladu (ID3 algoritmus) vypočítejte confusion matrix, Accuracy, Precision, Recall a F_1 skóre výsledného klasifikátoru.

6. Formálně definujte: Accuracy, Precision, Recall a F_1 skóre pro multi-class klasifikaci.

7. Uvažme následující confusion matrix:

Actual	**Predicted**		
	A	**B**	**C**
A	5	2	1
B	0	4	2
C	1	1	6

Vypočítejte Accuracy, Precision, Recall, F_1 skóre pro všechny třídy.

Součástí zkoušky může být i sestavení multi-class confusion matrix.

8. Uvažme binární klasifikátor do tříd $\{0, 1\}$, který vrací pravděpodobnost třídy 1 (tedy hodnotu z intervalu $[0, 1]$). Následující tabulka popisuje výsledek aplikace tohoto klasifikátoru na 12 vzorových příkladů:

Index	1	2	3	4	5	6	7	8	9	10	11	12
Actual	1	1	1	0	1	1	0	1	0	0	0	0
Predicted	.9	.89	.87	.81	.5	.48	.45	.21	.12	.1	.05	.04

Zde **Actual** je správná třída, **Predicted** je výstupní hodnota klasifikátoru.

- Nalezněte hodnotu thresholdu T takovou, že

$$\text{Recall}[T] \leq 0.6$$

- Rozhodněte a zdůvodněte, zda existuje threshold T takový, že

$$\text{Accuracy}[T] \geq 0.85$$

- Spočítejte kompletně ROC křivku.

  Specifikujte její hodnoty jako v přednášce, uvažujte threshold T ve vhodných intervalech.

9. Prezentujte (formálně) Bayesovský klasifikátor. Tedy kompletně popište algoritmus, který na vstupu dostane vektor vlastností klasifikovaného objektu a na výstupu dá třídu.

10. Uvažme dvě kategorie (categories) $\{\mathbf{1}, \mathbf{0}\}$ a dvě binární vlastnosti (features):

$$X_1 : \Omega \to \{a, b\}, X_2 : \Omega \to \{c, d\}$$

Máte k dispozici následující podmíněné pravděpodobnosti:

$$P(\mathbf{1}) = 0.4$$

$$P(X_1 = a, X_2 = c | Y = \mathbf{1}) = 0.1$$
$$P(X_1 = a, X_2 = d | Y = \mathbf{1}) = 0.2$$
$$P(X_1 = b, X_2 = c | Y = \mathbf{1}) = 0.3$$
$$P(X_1 = b, X_2 = d | Y = \mathbf{1}) = 0.4$$

(a) Klasifikujte (b, d) pomocí Bayesovského klasifikátoru (Bayes classifier) založeného na výše uvedených pravděpodobnostech a předpokladu, že $P(b, d) = 0.3$.

Pozor! Nejedná se o naivní Bayesovský klasifikátor, ale o plný Bayes.

(b) Je Bayesovský klasifikátor optimální?

11. Uvažme dvě kategorie (categories) $\{\mathbf{1}, \mathbf{0}\}$ a tři binární vlastnosti (features):

$$X_{color} : \Omega \to \{red, blue\}, X_{size} : \Omega \to \{large, small\}, X_{shape} : \Omega \to \{circle, square\}$$

Máte k dispozici následující podmíněné pravděpodobnosti:

$$P(\mathbf{1}) = 0.6$$

$$P(X_{color} = red|Y = \mathbf{1}) = 0.7$$
$$P(X_{color} = red|Y = \mathbf{0}) = 0.5$$

$$P(X_{size} = large|Y = \mathbf{1}) = 0.3$$
$$P(X_{size} = large|Y = \mathbf{0}) = 0.6$$

$$P(X_{shape} = circle|Y = \mathbf{1}) = 0.4$$
$$P(X_{shape} = circle|Y = \mathbf{0}) = 0.3$$

Pomocí naivního Bayesovského klasifikátoru (naive Bayes), založeného na výše uvedených pravděpodobnostech, klasifikujte $(blue, small)$, $(small, square)$ a $(blue, small, square)$. Nestačí pouze zapsat výsledek klasifikace, je nutné popsat celý postup výpočtu.

Zde jsou podmíněné pravděpodobnosti dány, ale můžeme i požadovat jejich odhad z tabulky (viz. přednáška).

12. Prezentujte (formálně) perceptronový učící algoritmus.

(Pozor, je nutné prezentovat i učící dataset!)

13. Prezentujte algoritmus gradientního sestupu (gradient descent) pro lineární regresi.

(Pozor, je nutné prezentovat i učící dataset a chybovou funkci (error function)!)

14. Demonstrujte kompletní výpočet perceptronového algoritmu na tréninkové množině

$$D = \{((1, 2), 1), ((2, 1), 0)\}$$

za předpokladu, že $\vec{w}^{(0)} = (0, -1, -1)$ a $\varepsilon = 1$. (Pamatujte, že $sgn(y) = 1$ pro $y \geq 0$ a $sgn(y) = 0$ pro $y < 0$.)

Poznámka: V podobném znění se na zkoušce může objevit lineární regrese!

15. Prezentujte formálně kompletní algoritmus K-means clustering.

16. Uvažme následující množinu hodnot:

$$\{0, 2, 3, 5, 8, 9, 10\}$$

Demonstrujte tři iterace algoritmu K-means clustering pro $K = 2$ a počáteční nastavení center na 4 a 7.

17. Uvažme následující tabulku vzdáleností objektů z množiny

$$\{A, B, C, D, E, F\}$$

	A	B	C	D	E	F
A	**0**	4	25	24	9	7
B	4	**0**	21	20	5	3
C	25	21	**0**	1	16	18
D	24	20	1	**0**	15	17
E	9	5	16	15	**0**	2
F	7	3	18	17	2	**0**

Demonstrujte kompletní postup aglomerativního hierarchického shlukování (agglomerative hierarchical clustering) pro všechny typy linkage: Single, complete, average. Výsledek znázorněte pomocí dendrogramů.

Uvažte jeden z výsledných clusteringů a vypočtěte silhouette score pro celou množinu objektů (je to průměr silhouette score pro jednotlivé objekty v množině).

18. Support vector machines.

    (a) Definujte pojmy support vectors a margin (zde postačí obrázek).

    (b) Formulujte výpočet SVM jako kvadratický optimalizační problém (quadratic optimization problem). Zde požadujeme přesnou formulaci včetně kompletního matematického zápisu!

V podobném znění se mohou objevit libovolné části teorie, kterýkoliv algoritmus.

19. Pro danou tréninkovou množinu $D = \{((0, -1), 1), ((2, 1), 1), ((6, -1), -1)\}$ nalezněte lineární model, který maximalizuje margin (tedy SVM). Model popište ve tvaru váhového vektoru (w_0, w_1, w_2).

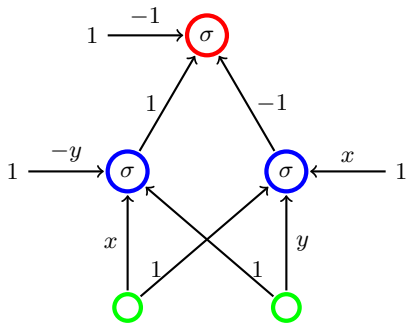
Zkuste si to vyřešit obrázkem i s použitím teorie z přednášky (řešte kvadratický program, uvědomte si, čemu v něm odpovídají „support vectors“). Pro důkladnější procvičení zkuste vyhodit první příklad z D, tedy pracovat jen s $D = \{((2, 1), 1), ((6, -1), -1)\}$ a také zkuste uvážit $D = \{((1, -1), 1), ((2, 1), 1), ((6, -1), -1)\}$.

20. Definujte pojem formálního neuonu.

21. Prezentujte formálně kompletní algoritmus zpětné propagace (backpropagation) pro vícevrstvé neuronové sítě.

    Pozor: Zkouška bude obsahovat alespoň jednu otázku vyžadující znalost a pochopení zpětné propagace!

22. Mějme následující dvouvrstvou síť 2-2-1 (tedy se dvěma vstupními, jedním výstupním a dvěma skrytými neurony):



    Zde

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq 0 \\ 0 & \xi < 0 \end{cases}$$

    je aktivační funkcí každého neuronu, x a y jsou proměnné nabývající libovolných reálných hodnot.

    (a) Demonstrujte kompletní vyhodnocení sítě pro vstup $(1, 1)$ při hodnotách proměnných $x = 3$ a $y = 2$.

    (b) Nalezněte x, y takové, že pro vstup $(1, -2)$ je výstup sítě roven 0.

    (c) Popište množinu všech možných přiřazení hodnot proměnným x, y takových, že pro vstup $(1, -2)$ je výstup sítě 1.

    (Nápověda: Přiřazení hodnot proměnným x, y lze zapsat jako dvojici reálných čísel, stačí tedy popsat příslušnou množinu dvojic pomocí soustav nerovnic.)

23. Dejte příklad neuronové sítě se třemi vstupy, jedním výstupem a aktivační funkcí

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq 0 \\ 0 & \xi < 0 \end{cases}$$

    která počítá funkci $F : \{0, 1\}^3 \to \{0, 1\}$ splňující následující (uvažujte každou podmínku zvlášť):

    (a) $F(x, y, z) = 1$ pro všechna $x, y, z \in \{0, 1\}$

(b) $F(x, y, z) = 1$ právě tehdy, když $(x + y) \cdot z >= 1$

(c) $F(x, y, z) = 1 - F(y, x, 1 - z)$ pro všechna $x, y, z \in \{0, 1\}$ taková, že $x \neq y$

(Pamatujte, že chování sítě nás zajímá pouze na vstupech z $\{0, 1\}^3$.)

Zamyslete se nad tím, co umí jeden neuron. Potom co dokážete dělat pomocí skládání neuronů a užitím logických/množinových operací (které lze opět implementovat pomocí neuronů ve vyšších vrstvách). Příkladů jako je tento lze vymyslet nekonečně mnoho :-).

24. Dejte příklad neuronové sítě se dvěma vstupy, jedním výstupem a

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq 0 \\ 0 & \xi < 0 \end{cases}$$

která počítá funkci $F : \mathbb{R}^2 \to \{0, 1\}$ splňující následující (uvažujte každou podmínku zvlášť):

- $F(x, y) = 1$ právě tehdy, když (x, y) patří do trojúhelníku s vrcholy $(0, 0)$, $(1, 0)$, $(0, 2)$
- $F(x, y) = 1$ právě tehdy, když (x, y) patří buď do trojúhelníku s vrcholy $(0, 0)$, $(1, 0)$, $(0, 2)$ nebo do trojúhelníku s vrcholy $(0, 0)$, $(-1, 0)$, $(0, -2)$

Pozor! Zde nás zajímá chování na všech vektorech $(x, y) \in \mathbb{R}^2$.

Řešte podobně jako předchozí příklad, pouze si uvědomte, že teď záleží na každém bodu v rovině.

25. Uvažmě množinu bodů v rovině

$$\{(0, 0), (1, 0), (0, 1), (4, 3)\}$$

Spočítejte local outlier factor, LOF_k, pro všechny čtyři body (objekty) a $k = 2$. Jak to dopadne pro $k = 1$? Jak pro $k = 4$?

Můžete zaokrouhlovat na dvě desetinná místa.