Algorithms of Information Security: Malware - exercises

Faculty of Information Technology
Czech Technical University in Prague

November 25, 2020

The k-nearest neighbors classifier

- The k-nearest neighbors (KNN) classifier is one of the most popular supervised learning methods.
- Let $T = \{(x_1, c_1), \ldots, (x_m, c_m)\}$ be the training set, where x_i is training vector and c_i is the corresponding class label.
- Given a query point x_q, its unknown class c_q is determined as follows:
  1. Find the set $T' = \{(x_1, c_1), \ldots, (x_k, c_k)\}$ of k nearest neighbors to the query point x_q.
  2. Then assign the class label c_q to the query point x_q by majority vote of its nearest neighbors.

The k-nearest neighbors classifier

- Majority vote is defined as:

$$c_q = \arg\max_{c} \sum_{(x_i, c_i) \in T'} \delta(c, c_i), \tag{1}$$

where c is a class label, c_i is the class label for $i-$th neighbor among k nearest neighbors of the query point, and $\delta(c, c_i)$ takes a value of one if $c = c_i$ and zero otherwise.

Distance-weighted k-nearest neighbor procedure

- Distance-weighted k-nearest neighbor procedure (WKNN) was first introduced as an improvement to KNN (in 1976)
- This extension is based on the idea that closer neighbors are weighted more heavily than such neighbors that are far away from the query point.
- KNN implicitly assumes that all k nearest neighbors are equally important in making a classification decision, regardless of their distances to the query point.
- In WKNN, nearest neighbors are weighted according to their distances to the query point as follows.

Distance-weighted k-nearest neighbor procedure

- Let $x_1, \dots, x_k$ be k nearest neighbors of the query object and $d_1, \dots, d_k$ the corresponding distances arranged in increasing order.
- The weight w_i for i-th nearest neighbor is defined as:

$$w_i = \begin{cases} \frac{d_k - d_i}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{otherwise} \end{cases}$$

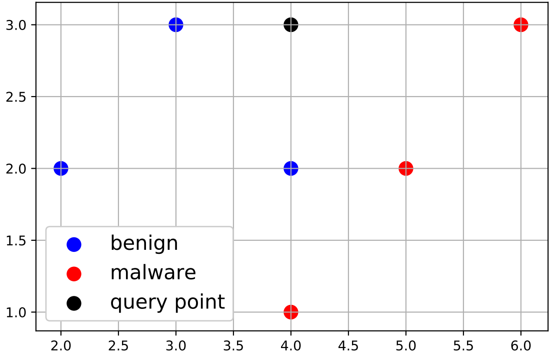
- The resulting class of the query point is then defined by the majority weighted vote as follows:

$$c_q = \arg\max_c \sum_{(x_i, c_i) \in T'} w_i \cdot \delta(c, c_i). \tag{2}$$

KNN - exercise 1

- Let training set $T = \{((2,2),\mathcal{C}),((3,3),\mathcal{C}),((4,2),\mathcal{C}),$ $((4,1),\mathcal{M}),((5,2),\mathcal{M}),((6,3),\mathcal{M})\}$
- Let $x_q = (4,3)$ and $k = 3$.
- Use KNN classifier and determine c_q.

KNN - solution 1



- $T' = \{((3,3), \mathcal{C}), ((4,2), \mathcal{C}), ((5,2), \mathcal{M})\}$ is set of $k = 3$ nearest neighbors.
- Since the majority class is benign then $c_q = \mathcal{C}$.

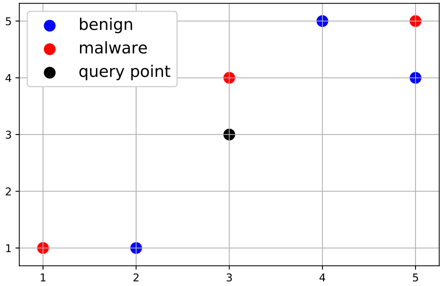
KNN - exercise 2

- Let training set $T = \{((2,3), \mathcal{C}), ((3,3), \mathcal{C}), ((3,2), \mathcal{C}), ((0,1), \mathcal{M}), ((1,0), \mathcal{M}), ((0,0), \mathcal{M})\}$
- Let $x_q = (2,1)$ and $k = 3$.
- Use KNN classifier and determine c_q.

WKNN - exercise 1

- Let training set $T = \{((2,1),\mathcal{C}),((4,5),\mathcal{C}),((5,4),\mathcal{C}),$ $((1,1),\mathcal{M}),((3,4),\mathcal{M}),((5,5),\mathcal{M})\}$
- Let $x_q = (3,3)$ and $k = 3$.
- Use WKNN classifier and determine c_q.

WKNN - solution 1



- $T' = \{((3,4),\mathcal{M}),((4,5),\mathcal{C}),((5,4),\mathcal{C})\}$ is set of $k=3$ nearest neighbors.
- Weight $w_i = \frac{d_k - d_i}{d_k - d_1}$, so $w_1 = \frac{\sqrt{5}-1}{\sqrt{5}-1} = 1$, $w_2 = \frac{\sqrt{5}-\sqrt{5}}{\sqrt{5}-1} = w_3 = 0$
- Since $1 > 0 + 0$ then $c_q = \mathcal{M}$.

WKNN - exercise 2

- Let training set $T = \{((2,2),\mathcal{C}),((2,3),\mathcal{C}),((3,3),\mathcal{C}),((0,0),\mathcal{M}),((0,1),\mathcal{M}),((2,0),\mathcal{M})\}$
- Let $x_q = (2,1)$ and $k = 3$.
- Use WKNN classifier and determine c_q.

Naive Bayes

- We present Naive Bayes for binary (two-class) classification problem.
- A Naive Bayes classifier is a probabilistic algorithm based on Bayes' Theorem that predicts the class with the highest *a posteriori* probability.
- Bayes theorem states that:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad (P(B) \neq 0).$$

- Assume a set of two classes $\{\mathcal{C}, \mathcal{M}\}$, where $\mathcal{C}$ denotes the class of benign samples and $\mathcal{M}$ denotes the class of malware.

Naive Bayes

- Training datasets are provided and a new (unknown) sample, which is represented by a feature vector $x = (x_1, \dots, x_n)$, is presented.
- Let $P(\mathcal{M}|x)$ denote the probability that a sample is malicious given the feature vector x that describes the sample. Similarly, $P(\mathcal{C}|x)$ denotes the probability that a sample is benign given the feature vector x that represents the sample. The Naive Bayes classification rule is stated as

$$\begin{aligned}\text{If } P(\mathcal{M}|x) &< P(\mathcal{C}|x),\ x \text{ is classified as benign sample} \\ \text{If } P(\mathcal{M}|x) &> P(\mathcal{C}|x),\ x \text{ is classified as malware} \end{aligned} \tag{3}$$

Naive Bayes

- The *a posteriori* probabilities $P(C|x)$ may be expressed in terms of the *a priori* probabilities and the $P(x|C)$ probabilities using Bayes' theorem as

$$P(C|x) = \frac{P(x|C)\ P(C)}{P(x)} \tag{4}$$

- Assuming that the values of the attributes (features) are conditionally independent of one another, the equation (4) may be expressed as

$$P(C|x) = \frac{\prod_{i=1}^{n} P(x_i|C)\ P(C)}{P(x)} \tag{5}$$

Naive Bayes

- Probabilities $P(x_i|C)$ can be estimated from the training set by counting the attribute values for each class.
- More precisely, the probability $P(x_i = h|C)$ is represented as the number of samples of class C in the training set having the value h for attribute x_i, divided by the number of samples of class C in the training set.
- The output of the classifier is the highest probability class C' :

$$C' = \arg\max_{C} \left(P(C) \prod_{i=1}^{n} P(x_i|C) \right) \tag{6}$$

Naive Bayes - exercise 1

- Let training set $T = \{((a,a,b),\mathcal{C}),((a,b,a),\mathcal{C}),((b,a,a),\mathcal{C}),$ $((a,b,b),\mathcal{M}),((b,a,b),\mathcal{M}),((b,b,a),\mathcal{M})\}$
- Let $x_q = x = (b,b,b)$
- Use Naive Bayes classifier and determine c_q.

Naive Bayes - solution 1

- We need to compute $P(\mathcal{M}|x)$ and $P(\mathcal{C}|x)$
- Bayes' theorem states that

$$P(C|x) = \frac{P(x|C)\ P(C)}{P(x)},$$

where $C \in \{\mathcal{M}, \mathcal{C}\}$.

- Assuming that the values of the features are conditionally independent of one another, then

$$P(C|x) = \frac{\prod_{i=1}^{3} P(x_i|C)\ P(C)}{P(x)}$$

- The denominator $P(x)$ can be omitted since it does not depend on the class C.

Naive Bayes - solution 1

- For each $i = 1, 2, 3$ and for each $C \in \{\mathcal{M}, \mathcal{C}\}$ we need to compute (using T) $P(x_i = b|C)$, and also $P(C)$:
- $P(x_1 = b|\mathcal{M}) = \frac{2}{3}, P(x_2 = b|\mathcal{M}) = \frac{2}{3}, P(x_3 = b|\mathcal{M}) = \frac{2}{3}$
- $P(x_1 = b|\mathcal{C}) = \frac{1}{3}, P(x_2 = b|\mathcal{C}) = \frac{1}{3}, P(x_3 = b|\mathcal{C}) = \frac{1}{3}$
- $P(\mathcal{C}) = \frac{3}{6} = \frac{1}{2} = P(\mathcal{M})$

Naive Bayes - solution 1

- $P(\mathcal{C}|x) \rightarrow P(x_1 = b|\mathcal{C}) \cdot P(x_2 = b|\mathcal{C}) \cdot P(x_3 = b|\mathcal{C}) \cdot P(\mathcal{C}) =$
  $= \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{54}$
- $P(\mathcal{M}|x) \rightarrow P(x_1 = b|\mathcal{M}) \cdot P(x_2 = b|\mathcal{M}) \cdot P(x_3 = b|\mathcal{M}) \cdot$
  $\cdot P(\mathcal{M}) = \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{2} = \frac{8}{54}$
- Since $\frac{8}{54} > \frac{1}{54}$, then x is classified as malware, i.e. $c_q = \mathcal{M}$.

Naive Bayes - exercise 2

- Let training set $T = \{((a,a,b),\mathcal{C}),((a,b,a),\mathcal{C}),((b,a,a),\mathcal{C}),((a,b,b),\mathcal{M}),((b,a,b),\mathcal{M}),((b,b,a),\mathcal{M})\}$
- Let $x_q = (a,a,a)$
- Use Naive Bayes classifier and determine c_q.