Johannes Gutenberg-Universität Mainz
Institut für Informatik
Data Mining
Prof. Dr. Stefan Kramer

Machine Learning - Sheet 2

04.05.2017
Deadline: 11.05.2017 - 10:00

Task 1: Decision Tree *(9 Points)*

We now finish the implementation of the basic decision tree algorithm, as described in Section 3.4 (page 55) in [2] (`Exercise_02_01.java`).

(a) Implement the data structure of the decision tree (inner nodes, leafs, the actual decision method), provide two factory methods `branch` and `leaf`.

(b) Implement a method or function that performs the attribute selection for a given node (reuse `informationGain` method already implemented in a previous exercise).

(c) Implement methods `trainModelOnSubset` and `trainModel`.

(d) Test your implementation on the Weather dataset (`weather.nominal.arff`).

(e) Evaluate your decision tree using the car dataset (`http://archive.ics.uci.edu/ml/datasets/Car+Evaluation`) and a very simple procedure: Randomly split the dataset into a training set (two thirds) and a test set (one third) (`Exercise_02_01#splitTrainTest`). Take the training set to train your decision tree. Afterwards, compute the percentage of correctly classified instances from the test set (`Exercise_02_01#evaluate`).

Task 2: Boosting *(5 Points)*

In order to understand how boosting works, we will use `http://scikit-learn.org`. Make yourself familar with AdaBoost example available on `http://scikit-learn.org/stable/auto_examples/ensemble/plot_adaboost_twoclass.html`. You do not need to understand every detail of the given Python script, but you should have a basic understanding of what is happening. Change the parameter `n_estimator` in

```
AdaBoostClassifier(DecisionTreeClassifier(max_depth=1),
                   algorithm="SAMME",
                   n_estimators=200)
```

to different values (e.g., $1, 2, 5, 10, 20, 30, 40, 50$) and compare the plots. Discuss the main idea of AdaBoost using at least three of these plots.

Task 3: Bootstrap *(6 Points)*

We want to understand where the number "0.632" in "0.632-Bootstrap" comes from. We want to derive an exact formula, and compare it with the output of a random experiment (`Exercise_02_03.java`).

(a) Read the first three paragraphs in subsection "8.5.4 Bootstrap" (page 371) in "Data Mining" by Han et al. [1]

(b) Implement the `bootstrap` sampling method (`Exercise_02_03#bootstrap`).

(c) Use the `bootstrap` method to implement the following random experiment (`Exercise_02_03#randomProportionDrawn`):

- Create a set with d distinct integers
- Draw (with replacement) $k \cdot d$ samples from the set (where k is a positive integer factor)
- Compute the proportion of the instances that have been drawn.

(d) Run the randomized experiment with some sufficiently large d (e.g. $d = 10000000$) and $k = 1$. What do you observe? *(in written form, or as a comment in nearby code)*

(e) Give an exact formula for the expected proportion of instances that are drawn if we draw $k \cdot d$ samples with replacement (`Exercise_02_03#expectedProportionDrawn`). Hints:

- Consider an arbitrary but fixed instance
- What is the probability that this instance is drawn, if we draw a single sample?
- What is the probability that this instance is *not* drawn?
- What is the probability that this instance is *not* drawn, if we repeatedly draw $k \cdot d$ samples (uniformly, independently, with replacement)?

(f) What happens with the expected value if the size d of the dataset becomes very large? (`Exercise_02_03#expectedProportionDrawnInLimit`)
Hints:

- Compute the limit for $d \to \infty$ of the `expectedProportionDrawn` formula.
- Make the substitution $n := k \cdot d$, use the fact that $\lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n = \exp(x)$.

(g) Compare the outputs of `randomProportionDrawn`, `expectedProportionDrawn`, and `expectedProportionDrawnInLimit` for various values of k and d, for example for $k \in \{1, 2, 3, 5, 10\}$, $d \in \{10, 100, 1000, 1000000, 100000000\}$. Briefly describe your observations *(in written form, or as comment in nearby code)*.

References

[1] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[2] Tom M. Mitchell. *Machine learning.* McGraw Hill series in computer science. McGraw-Hill, 1997.