MADS-ML – Machine Learning

Classification

Prof. Dr. Stephan Doerfel

FACHHOCHSCHULE KIEL
University of Applied Sciences

Moodle (WiSe 2024/25)

Motivation

Is this a picture of a dog?

Motivation

Is this a picture of a dog?

Motivation

Is this a picture of a dog?

Outline

The classification problem

Definition 1 (The classification problem (abstract))

Setting:

- a universe U of classifiable objects each described with features (attributes) from a common set of features A
- a set of classes C
- a set of labelled examples $O \subseteq U$ (meaning for $o \in O$, $c(o) \in C$ is known).

Task:

- Determine a **classification function** $K : U \to C$, that maps instances of U onto their respective class!

A classification function implies a partition on U with $|C|$ partitions. Each element belongs to exactly one partition.

Notebook 02_1_classification_digits_walkthrough, Cells 1–7

Labelled Data

labelled data: in practice, a labelled dataset consists of tuples (x_i, y_i), where $\mathrm{x}_i = (x_{i,1}, \ldots, x_{i,d})$ is a feature vector of length d and $y_i \in C$ is the target, i.e. the corresponding class of x_i.

features: The features can be

- numerical (length, volume, age, pixel values, . . .) or
- categorical (color, type, true / false, . . .).

Example: Digits For a data instance x, x_i is a vector of length $p = 64$ with numerical values. The classes are the digits $0, 1, \ldots, 9$.

Notebook 02_1_classification_digits_walkthrough, Cells 8–9

The classification problem

For non-trivial classification problems, finding a classification function is impossible or extremely difficult.

Definition 2 (Simplified classification problem)

Instead of finding the actual classification function, determine an approximation that minimizes some quality function assessing how close the approximation is to the actual classification function.

Remarks

- In a classification setting, the classes are known **apriori** (beforehand) – other than in clustering, where the classes have to be determined.
- Distinguishing classes is non-trivial, but actually not part of the classification problem.
- **Explainability** of the learned classifier is important and desirable. Such white-box classifiers (in contrast to black-box) allow understanding the decision.
- Usually, the classification must be fast (**online** performance), while the training should be fast (**offline** performance) and require only small training datasets.
- Creating proper features (e.g. turning numerical features into categorial ones, scaling) is part of the preparation of setting up a classifier.

Toy-Example: Classifying insurance risks

Lets consider the following training dataset of a (simplified) insurance company

ID	age	car type	risk (target)
1	23	family	high
2	17	family	high
3	43	sports	high
4	68	family	low
5	32	truck	low

One possible classification function is:

- if age $>$ 50 then risk $=$low;
- if age $\leq$ 50 and car type $=$ truck then risk $=$ low;
- if age $\leq$ 50 and car type $\neq$ truck then risk $=$ high.

Two phases of a classifier

Offline Phase – Learning

ID	age	car type	risk (target)
1	23	family	high
2	17	family	high
3	43	sports	high
4	68	family	low
5	32	truck	low

learning
training

yields

classifier

training data

classification-algorithm

Online Phase – Classification

unknown data → consumes → classifier → yields → classification

Outline

Evaluation

Situation:

How do we measure the quality of a trained classfier?

Evaluation data:

Use labelled data, feed it (without the label) to the classifier, test wether the classifier yields the correct class.

Data Source:

- WRONG: Use the same data as for training. Drawback: The algorithm already knows this data and thus the task's solution. ⌛ Details in Classification II ...
  NEVER evaluate predictive performance on training data!
- CORRECT: Before training, split the labelled data into training and test data. Use only the training dataset for the learning phase. Use the test dataset for the evaluation.

Notebook 02_1_classification_digits_walkthrough, Cells 10–14

Quality Measures for Classifiers

Let K be a classifier, TR the training set, TE the test set and $C(x)$ the correct class of a data instance x.

- classification accuracy:

$$A_{TE}(K) := \frac{|\{x \in TE \mid K(x) = C(x)\}|}{|TE|}$$

- true classification error:

$$E_{TE}(K) := \frac{|\{x \in TE \mid K(x) \neq C(x)\}|}{|TE|} = 1 - A_{TE}(K)$$

- apparent classification error:

$$E_{TR}(K) = \frac{|\{x \in TR \mid K(x) \neq C(x)\}|}{|TR|}$$

Note, that true and apparent classification error use the same formula on different data sets.

Apparent Classification Error

The apparent classification error is taken on the training data.

- It is not a suitable measure for classification quality!
- It is however interesting to compare to the true classification error
- the comparison informs on overfitting (⏳ tbd.)

Interpreting the quality measure

- compare algorithms, or different parametrizations of the same algorithm
- decide which is best (according to the quality function)
- But how much is good?
- Typical statement in a report on ML experiments: "The results show that our method is 300 times more successful than random baseline"

Minimum requirement: A classifier should at least outperform random guessing.

Is **58.5%** accuracy a good result?
Is **99.9%** accuracy a good result?
➔ We cannot judge such a result without a frame of reference!

Interpreting the quality measure

- compare algorithms, or different parametrizations of the same algorithm
- decide which is best (according to the quality function)
- But how much is good?
- Typical statement in a report on ML experiments: "The results show that our method is 300 times more successful than random baseline"

Minimum requirement: A classifier should at least outperform random guessing.

Guessing Baselines

Let C be the set of all classes and $c_{max} \in C$ the largest class (the one with the most data instances in the training dataset):

uniform distribution guessing baseline:

Pick a class at random where each class has the same probability: Result

$$A_{TE}(K) = \frac{1}{|C|}$$

largest class baseline:

Always "guess" the largest class c_{max}. Result:

$$A_{TE}(K) = \frac{|\{x \in TE \mid K(x) = c_{max}\}|}{|TE|}$$

Notebook 02_1_classification_digits_walkthrough, Cells 15

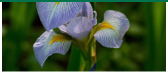
Exercises 1–2

Outline

Example Dataset

- Iris dataset
- often used in benchmarking classification algorithms
- instances are plants of genus iris
- features: 4 measures of petals and sepals (breadth and width)
- now with classes: 3 species (Iris Setosa, Iris Versicolor, Iris Virginica)

For this lecture, we mostly ignore the domain (in real applications, we don't!).



[1]

[2]

[3]

Referenzen

E. Hunt.
Iris virginica 2.jpg, 2018.
https://creativecommons.org/licenses/by-sa/4.0/.

Radomil.
Kosaciec szczecinkowaty Iris setosa.jpg, 2015.
https://creativecommons.org/licenses/by-sa/3.0/.

D. G. E. Robertson.
Blue Flag, Ottawa.jpg, 2005.
https://creativecommons.org/licenses/by-sa/3.0/.