University of Applied Sciences RheinMain
Prof. Dr. Adrian Ulges

20.10.2016

Machine Learning

Course Work 1

to complete by: 27.10.2016

Please execute this course work (as well as any following) in teams of two.

Exercise 1.1 (Review: Theory)
Review the following concepts, or get familiar with them (in case they are new to you):

a) **Calculus**
partial derivatives, the chain rule, gradient descent

b) **Descriptive Statistics**
expectation value, variance, covariance, correlation

c) **Probability Theory**
probability density functions, the law of total probability, Bayes' Rule, conditional probabilities, statistical independence

d) **Parameter Estimation**
maximum-likelihood (ML)

e) **Linear Algebra**
hyperplane representations (incl. the Hessian normal form), eigenvectors and -values, rotation matrices, matrix transposing and inversion, matrix multiplication.

Exercise 1.2 (Review: Python)
In case you are new to Python: Start getting familiar with it. The course homepage has some nice **links** to get you started. *Remark: Version 2.7 is pre-installed on the pool machines (including numpy, sklearn, matplotlib, and pandas).*

Exercise 1.3 (Review: Numpy, Pandas)
Before starting with hands-on work, have a look at Numpy and Pandas:

- The Numpy Quickstart Tutorial:
`https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`
- The '10 Minutes to Pandas':
`http://pandas.pydata.org/pandas-docs/stable/10min.html`

Exercise 1.4 (The Titanic)
A nice (though a bit morbid) beginner ML problem is the "Titanic": Given some data about the Titanic's passangers (like age, gender, or passenger class), predict who survived the disaster.

- Go to `kaggle.com` *(a website for machine learning competitions)* and search for the "Titanic" competition. Enroll for the competition and download the files `train.csv` and `test.csv`.
- Download `classifier_simple.py` from the course website and run the classifer. Check the code thoroughly (particularly, what does `map()` do?). Submit your result to Kaggle and check your accuracy score.
- Get creative: Inspect the data and hand-craft your own decision rule for survival prediction. Your rule should involve at least three features. *Hints: (1) You might find pandas'* `pivot_table()` *useful to check which features are promising. (2) Think about turning numerical features into categorial ones by thresholding: For example, split people into 'children' and 'grown-ups' by age.*
- Submit your result to Kaggle and get your accuracy checked. Did you improve?

Exercise 1.5 (Your Titanic Machine Learner ♡)
Train your own decision tree classifier for the Titanic problem, using `sklearn`.

- Check out *sklearn*'s class `DecisionTreeClassifier`: `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`
- Train your decision tree on `train.csv` and predict the survival rate on `test.csv`. Leave all parameters unchanged. *Remarks: (1) You may drop features that you obviously don't find useful (like the name). (2) sklearn's DecisionTreeClassifier only accepts numerical inputs (no strings like 'male'). Check out pandas'* `get_dummies()` *to turn categorial features into numerical ones.*
- For those familiar with Python: Optimize the `max_depth` parameter. Test the values 1,2,3,4 and 5 and run a 5-cross-validation on `train.csv` (with sklearn's `cross_val_score()` method, that is really simple).
- Submit your result to Kaggle and get your accuracy checked. Did you improve?

Exercise 1.6 (Report)
Put together a presentation of 2-3 slides, summarizing your hand-crafted model, your evaluation results on kaggle, and any open questions.