Ullrich Köthe, Manuel Haussmann
ullrich.koethe@iwr.uni-heidelberg.de

Exercise 4a

Deadline: 5.12.2018

Regulations

Provide your comments for last week's homework in the files `tree-methods-commented.ipynb` and `tree-methods-commented.html`. Hand-in your solutions to this week's tasks in notebooks `precision-recall.ipynb/precision-recall.html` and `red-cards.ipynb/red-cards.html`. Zip all files into a single archive with naming convention (sorted alphabetically by last names)

`lastname1-firstname1_lastname2-firstname2_exercise04a.zip`

or (if you work in a team of three)

`lastname1-firstname1_lastname2-firstname2_lastname3-firstname3_exercise04a.zip`

and upload it to Moodle before the given deadline.

Recall that we will give zero points if your zip-file does not conform to this naming convention.

1 Comment on your solution to exercise 4

Study the sample solution `ex04-solution.ipynb` provided on Moodle and use it to comment on your own solution to this exercise. Specifically, copy your original notebook `tree-methods.ipynb` to `tree-methods-commented.ipynb` and export it to `tree-methods-commented.html` in the end. Insert comments as markdown cells starting with

```
<span style="color:green;font-weight:bold">Comment</span>
```

in order to clearly distinguish your comments from other cell types. The point of these comments is that you identify your errors and bugs yourselves, so that you learn from your mistakes. In addition, the tutor will have an easier time distinguishing between the first mistake and consequential errors caused by the first one and will only deduct points for the former. If you fail to hand in comments, the tutor is not required to make this distinction and will deduct points for all errors alike.

2 Precision-Recall Curves

In this exercise, we use sklearn's digits dataset for an image retrieval experiment. Given any image from this dataset as a query, your algorithm shall find all similar images, where we define *similarity* by "contains the same digit". Of course, only the features (i.e. the pixel values of the images) may be used for similarity search. The known labels only serve for testing the quality of the search result.

2.1 Euclidean Distance (7 points)

Define dissimilarity by the Euclidean distance between pixel values

$$d(X_i, X_{i'}) = \|X_i - X_{i'}\|_2^2$$

To efficiently compute these distances, you should use vectorization (remember exercise 1b). Let D be the full dissimilarity matrix, i.e. $D_{ii'} = d(X_i, X_{i'})$. An `np.argsort()` of row D_i now gives the similarity ordering of all digits relative to query digit X_i. The response sets S_{im} consist of the m nearest instances to query X_i. The positive class is defined by the instances having the same label

as the query, i.e. $Y_{i'} = Y_i$, and $N_i = \#\{i' \in 1, ..., N : Y_{i'} = Y_i\}$ is the total number of positives. Each response set defines a pair $(\text{precision}_{im}, \text{recall}_{im})$ as

$$\text{precision}_{im} = \frac{\text{TP}_i(m)}{\text{TP}_i(m) + \text{FP}_i(m)} \qquad \text{recall}_{im} = \frac{\text{TP}_i(m)}{N_i}$$

where $\text{TP}_i(m) = \#\{i' \in S_{im} : Y_i = Y_{i'}\}$ and $\text{FP}_i(m) = \#\{i' \in S_{im} : Y_i \neq Y_{i'}\}$ are the number of true and false positives in S_{im}. Compute the precision matrix P and recall matrix R whose elements are the precision resp. recall values from these pairs (vectorization again helps). For each digit class $k \in \{0, ..., 9\}$, compute $\bar{P}_k$ and $\bar{R}_k$ as the average of the rows of P and R referring to class k, i.e. where $Y_i = k$. Plot the resulting precision/recall curves (using m as the free parameter) and determine the area-under-curve (AUC) for each k. Do not use `sklearn` in this task.

Repeat the same steps with precision gain and recall gain defined as

$$\text{precisionGain}_{im} = \frac{\frac{1}{\text{precision}_{im}} - \frac{N}{N_i}}{1 - \frac{N}{N_i}} \qquad \text{recallGain}_{im} = \frac{\frac{1}{\text{recall}_{im}} - \frac{N}{N_i}}{1 - \frac{N}{N_i}}$$

and comment on the differences.

2.2 Hand-Crafted Distance (7 points)

Try to improve the area-under-curve by defining a 2-dimensional feature space optimized for similarity search. You can compute the new features from the original pixel values in any way you want. Create a scatterplot of the resulting 2D dataset. Which property should this scatterplot have in order for the new features to be especially suitable for similarity search?

Repeat the experiment from 2.1 with the new features and comment on your results. If you cannot come up with features that improve the AUC, report results for the best features you found.

3 Red Cards Study

In this exercise, you will take a look at a recent experiment in crowdsourcing research. 29 teams of researchers were given the same dataset and the same question: *"Are football (soccer) referees more likely to give red cards to players with dark skin than to players with light skin?"*. Interestingly, all 29 teams arrived at different conclusions (finding no bias or slight bias or severe bias in referee decisions), despite having identical data and instructions. Read `http://www.nature.com/news/crowdsourced-research-many-hands-make-tight-work-1.18508` for a comment in the Nature journal by Raphael Silberzahn & Eric L. Uhlmann, the initiators of the experiment.

We ask you the same question: Given the dataset, can you confirm or refute the question? To do this, please download the dataset from `https://osf.io/gvm2z/` (`1. Crowdsourcing Dataset July 01, 2014 Incl.Ref Country.zip` contains the dataset, `README.txt` a detailed description of the data and `2._Crowdstorming_Pictures_Skin_Color_Ratings.zip` the images of the players[1]). Feel free to look at `Crowdsourcing Analytics - Final Manuscript.pdf` for a more detailed description of the experiment, its features, and the different methods participants applied to tackle the question.

3.1 Loading and Cleaning the Data (10 points)

The first step consists of loading the .csv file and preparing the data for the experiment. One participant of the official experiment provided a nice jupyter notebook demonstrating how the python library `pandas` can be utilized to achieve this: `http://nbviewer.ipython.org/github/mathewzilla/redcard/blob/master/Crowdstorming_visualisation.ipynb`. You should get inspiration from

[1] in case you want to improve skin color ratings on your own

this example, but still choose your own data preparation steps. The following questions may guide you:

- What do the feature names (e.g. column `games`) stand for?
- Which irrelevant features might be dropped?
- What relevant features might be missing, but can be computed? E.g., you can obtain the age of a player (which might be relevant) from his birthday, or create entirely new features by non-linear combinations of existing ones.
- Are there missing data values (e.g. missing skin color ratings), and how should they be dealt with? (see `https://en.wikipedia.org/wiki/Missing_data`)
- How good are the skin color ratings? Do the raters agree?
- Should referees with very few appearances be excluded from the dataset?
- Should features be normalized and/or centralized?

Categorical features (e.g. `league`) should be transformed to a one-hot encoding (see `https://en.wikipedia.org/wiki/One-hot`). In case of `league`, you can also repeat the experiment independently for the different leagues to check if there are differences between countries. Provide a detailed description and justification of your data preparation.

3.2 Model Creation (8 points)

Given features X_i of player i, we want to predict $Y_i = N_{i,\text{red}}/N_i$, the fraction of games where the player will receive a red card. We will solve this problem using two model types: linear regression and regression forests.

Linear regression determines a weighted sum of the features $\hat{Y}_i = X_i\hat{\beta} + \hat{b}$, where optimal weights and intercept minimize the squared error:

$$\hat{\beta}, \hat{b} = \text{argmin}_{\beta,b} \sum_i (X_i\beta + b - Y_i^*)^2$$

A regression forests works similarly to a decision forest (reuse your code from exercise 4), but leaf responses and split criteria differ:

- The response of leaf b_l is the average response of the training instances assigned to this leaf:

$$\bar{Y}_l = \frac{1}{N_l} \sum_{i \in b_l} Y_i^*$$

- The optimal split into children b_λ and b_ρ minimizes the squared error:

$$\sum_{i \in b_\lambda} (Y_i^* - \bar{Y}_\lambda)^2 + \sum_{i \in b_\rho} (Y_i^* - \bar{Y}_\rho)^2$$

Moreover, the test `'not node_is_pure(node)'` makes no sense for regression trees and should be eliminated. The forest's reponse is the average response of its trees.

Implement both models and determine their squared test errors by means of cross-validation.

3.3 Answering the Research Question (8 points)

Now perform a *permutation test* to answer the research question. To this end, create 19 new training sets where the skin color variable is randomly shuffled among the players. Each dataset uses a different permutation of skin colors, but keeps all other features and the response intact. This

ensures that any possible association between skin colors and responses Y_i^* is destroyed, whereas the marginal skin color distribution gets preserved.

Determine the squared errors of the two model types on these new datasets by cross-validation as well. If all 19 datasets exhibit higher test errors than the original unscattered dataset, you can conclude that there is a skin color bias in red card decisions with a p-value of $p = 1/20 = 0.05$. If so, determine the direction of the bias by comparing the average of the Y_i^* for light and dark colored players.

Play with the data cleaning procedure with the following goal: Find two equally plausible cleaned datasets that give opposite answers to the research question, i.e. one uncovers a skin color bias, and the other does not. If you succeed in finding such datasets, it demonstrates how easy it is in practice to tweak the data in the direction of the desired outcome, and how careful one needs to be conducting statistical research and interpreting published results.

In any case, keep in mind that a statistical analysis like this can only reveal *correlations* between features and response, but says nothing about the direction of causality (statistical analysis of causality is also possible, but requires more powerful methods and larger datasets). Provide two alternative plausible causal hypotheses, besides the obvious "referees discriminate against dark colored players", that might explain a possible correlation.