General Q&A

No questions.

Theory Tasks

Chapter 1: Introduction

> Explain the difference between over- and underfitting!

- overfitting:
  - model is too complex for dataset
  - model also captures outliers/noise that are not generalizable
- underfitting:
  - model is too simple for dataset
  - model not able to capture general trend in data properly

> How can we use training data and validation/test data to avoid overfitting?

- can observe overfitting as difference in prediction performance between training data and validation/test data
- in particular, models might be better on data they are trained on than on separate data
- thus, should split data into training part and validation/test part
- train models on training data only
- make predictions and observe prediction performance mainly on validation/test data
- split methods like holdout split, cross-validation, etc. will be discussed in a later lecture

> How can we avoid overfitting when trying to decide which order of regression model to fit to our data?

- split data into training part and validation part
- train models of different order on training data
- compare models on validation data; pick the one with highest prediction performance
- this should result in best compromise between underfitting and overfitting
- in contrast, on training data, prediction performance probably just gets better the more complex the model gets

> Explain the One-Rule algorithm! Why is it used for small/noisy datasets?

- rough explanation:
  - check for each value of each feature which class is most frequent
  - count all data objects with deviating class labels as classification errors
  - count number of errors per feature
  - pick feature with lowest error
  - for numeric features, discretization necessary
- benefits: very simple model, less prone to overfitting than more complex models
  - makes it suitable for small/noisy datasets
  - main danger causing overfitting: features with many values
    * can also happen for categorical features, not only numeric ones

Chapter 2: Fundamentals

Data & Descriptive Statistics

How can you categorize data?

- basic categorization from lecture:
  - categorical
    - nominal
    - ordinal
  - numerical
    - discrete
    - continuous
- other categorizations possible, e.g., by dimensionality (part of lecture as well):
  - one-dimensional (univariate)
  - multi-dimensional (multivariate); special cases:
    - two-dimensional (bivariate)
    - high-dimensional (definition of 'high' depends on use case)
  - dimensionless

Compare the three types of aggregates!

- distributive: store only the desired aggregate for each partition of data
- algebraic: store multiple (but fixed number of) aggregates for each partition of data
- holistic: number of aggregates to be stored for each partition is potentially unlimited
  - i.e., in worst case, need to store all data objects, so no aggregation at all

Why are distributive aggregates efficient?

- need only to store one aggregate for each (arbitrarily large) partition of data
- from partition aggregates, can directly compute value of aggregate for full dataset
  - i.e., without needing to look into (individual data objects of) partitions again
- can also be easily updated if new data arrives:
  - i.e., self-maintainable regarding insertion
  - first, compute aggregate on new data
  - second, combine with aggregate on old data to get overall value of aggregate

What is the difference between variance and covariance?

- variance
  - univariate statistic
  - describes how much a random variable varies around its expected value (i.e., mean)
  - $Var(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$ (population variance)
- covariance
  - bivariate statistic
  - describes how much two random variables vary in same direction
  - $Cov(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$
  - unnormalized (gets larger if you scale variables larger, e.g., multiply with factor)
    - Pearson correlation is a normalized version of this statistic

Statistical Tests

What is the null hypothesis of the χ^2 test?

- in lecture: two random variables are independent
- more general: check frequencies of one random variable against some distribution

> How can one conduct feature selection with the χ^2 test?

- see subtask f) of first programming-exercise sheet
- χ^2 test statistic collects evidence against independence of random variables
- i.e., it can be used as a measure of dependency
  - if you want a normalized measure, can use *1 - p-value* as well
- use case 1: only select features that show highest dependency to target variable
  - i.e., keep most relevant features
  - can be absolute number (select top k features)
  - can be threshold-based (select all features with a p-value lower than some threshold)
- use case 2: if two features show a strong dependency to each other, remove one of them
  - i.e., reduce redundancy in data

> Does the χ^2 test also work if random variables have a non-normal distribution?

- yes
- non-parametric test, i.e., does not assume any distribution
- also, works with categorical data (which are not normally distributed anyway)

> Assume two samples are given. Is the following statement correct? "If a Kolmogorov-Smirnov test concludes that both samples were not drawn from the same distribution, then a Wilcoxon-Mann-Whitney test is redundant."

- Wilcoxon-Mann-Whitney test compares distributions based on median
- KS test compares distributions based on whole cumulative distribution function
- thus, KS test seems more general and Wilcoxon-Mann-Whitney test seems redundant
- if it comes to actual statistical power, situation is more complicated, as paper Comparison of the Powers of the Kolmogorov-Smirnov Two-Sample Test and the Mann-Whitney Test for Different Kurtosis and Skewness Coefficients Using the Monte Carlo Simulation Method shows
- decision also depends on what you actually want to compare
  - in some situations, you are only interested in central tendency of distribution
  - i.e., spread of distribution, long tails, etc. might be irrelevant for you

Data Reduction

> Name three different ways to reduce data!

- numerosity reduction: reduce number of data objects
- dimensionality reduction: reduce number of features
- discretization: reduce number of values per feature

> How can you automatically select important features? What do you need to consider to perform feature selection?

- features should be relevant for prediction target
- features should not be redundant to each other
- search strategy for feature sets should be efficient
  - since there are 2^n feature sets for n features
  - some approaches evaluate each feature individually
    - e.g., how strongly is each feature correlated to target variable
  - some approaches iterate over space of potential feature sets
    - e.g., greedy forward/backward selection, genetic algorithms, etc.
  - some prediction models implicitly select features
    - e.g., decision trees

> How does PCA work?

- is a dimensionality-reduction technique
- rotation: does not select (original) features, but transforms dataset to new basis
- projection: under new basis, first few features should capture most variance of dataset
- mathematical details:
  - principal components are eigenvectors of covariance matrix of original dataset
    - computation based on covariance matrix is called *eigendecomposition*
    - also other ways of computation, e.g., via singular value decomposition (SVD)
  - principle components constitute columns of rotation matrix
    - form an orthonormal basis of feature space
    - transformation (rotation) is a simple matrix multiplication
    - features under new basis are linear combinations of features from original dataset
    - transformation from new basis back to old one is possible as well
  - eigenvalues associated with principal components (eigenvectors) help to project data
    - are proportional to amount of variance captured in principal component
    - components in rotation matrix should be ordered by decreasing eigenvalues
    - to reduce dimensions, only keep first few columns of rotation matrix
    - if you keep all components, transformation is lossless
  - features under new basis are uncorrelated
  - to conduct PCA, features should be standardized
    - mean of zero (if PCA computed via covariance matrix, this happens automatically)
    - variance of one (else, feature with higher variance dominate result)