R for text analysis: Supervised Machine Learning

Wouter van Atteveldt

April 2019
Overview

- What is Machine Learning?
- Principles and validation
- Algorithms
What is Machine Learning

- Form of Statistical Learning
- Given data, find a regularity to predict new data
- Supervised machine learning:
  - Build a model using labeled examples
  - Using many input features (independent variables)
  - To predict the output class (dependent variable)
Supervised Machine Learning

CLASSICAL MACHINE LEARNING

SUPERVISED

Classification
- Data is pre-categorized or numerical
- Predict a category
- "Divide the socks by color"

Regression
- Predict a number
- "Divide the ties by length"

UNSUPERVISED

Clustering
- Data is not labeled in any way
- Divide by similarity
- "Split up similar clothing into stacks"

Association
- Identify sequences
- "Find hidden dependencies"
- "Find what clothes I often wear together"

Dimension Reduction
- Find hidden dependencies
- "Make the best outfits from the given clothes"
Machine learning process
Machine learning vs "normal" modeling

• Same principle, different goal
• Classical modeling: explain / understand
  • Requires interpretable parameters
• Machine learning: prediction (aka forecasting)
  • Requires high accuracy
• So...
  • ML models use many more features
  • Which are often strong multicollinear
  • But that’s ok, since we don’t care about the parameters
How to prevent overfitting

- Data can always be modeled perfectly with a complex enough model
- How to prevent overfitting?
How to prevent overfitting

• Data can always be modeled perfectly with a complex enough model

• How to prevent overfitting?

• Estimation: 'Regularize' model (bias parameters towards zero)

• Validation: Use train-(develop)-test split
Machine Learning Algorithms

- Estimate model based on data
- Many algorithms exist
- All deal with data scarcity and overfitting problems in some way
- Most need some sort of (hyper)parameter tuning

- How to choose?
- Differences often not that big
- Can try all models/parameters on development set, pick the best
- Can even combine multiple models (ensemble methods)
Naive Bayes

\[
P(c | x) = \frac{P(x | c) P(c)}{P(x)}
\]

\[
P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \cdots \times P(x_n | c) \times P(c)
\]
Support Vector Machines

The diagram illustrates a Support Vector Machine (SVM) in a two-dimensional space. The SVM aims to find an optimal hyperplane that maximizes the margin between the two classes. The optimal hyperplane is the one that is as far as possible from the closest data points of each class. The margin is the distance between the hyperplane and the closest data points, known as support vectors.
Neural Networks (perceptron)
Neural Networks (multi-layer)
Text Mining: classical approach

- DFM is data
- Run ML model with frequencies as features
- Feature engineering to improve performance
  - e.g. selection, weighting, n-grams, NLP, dictionaries, ...
Text Mining: deep learning

- Deep learning = very large (structured) neural networks
- Include feature extraction / engineering in neural network
  - Word embeddings, convolutions, LSTMs, ...
  - Pre-train with unannotated data (embeddings)
  - Can use context (no bag-of-words assumption)
- Yoav Goldberg, *Neural Network Methods in Natural Language Processing*
There must be a package, right?
There must be a package, right?

- More like 100 of them
- See https://cran.r-project.org/web/views/MachineLearning.html
- Which ones to use?
There must be a package, right?

• More like 100 of them
• See https://cran.r-project.org/web/views/MachineLearning.html
• Which ones to use?
• Quanteda actually does naive bayes, scaling
• Single algorithms:
  • nnet, e1071, ...
• Comprehensive packages:
  • caret, CoreTools
Steps before running machine learning

- (May need to upgrade to R 3.5)
- Data cleaning, feature selection
- Split in train, test, (validation)
Running the model

(see handout!)