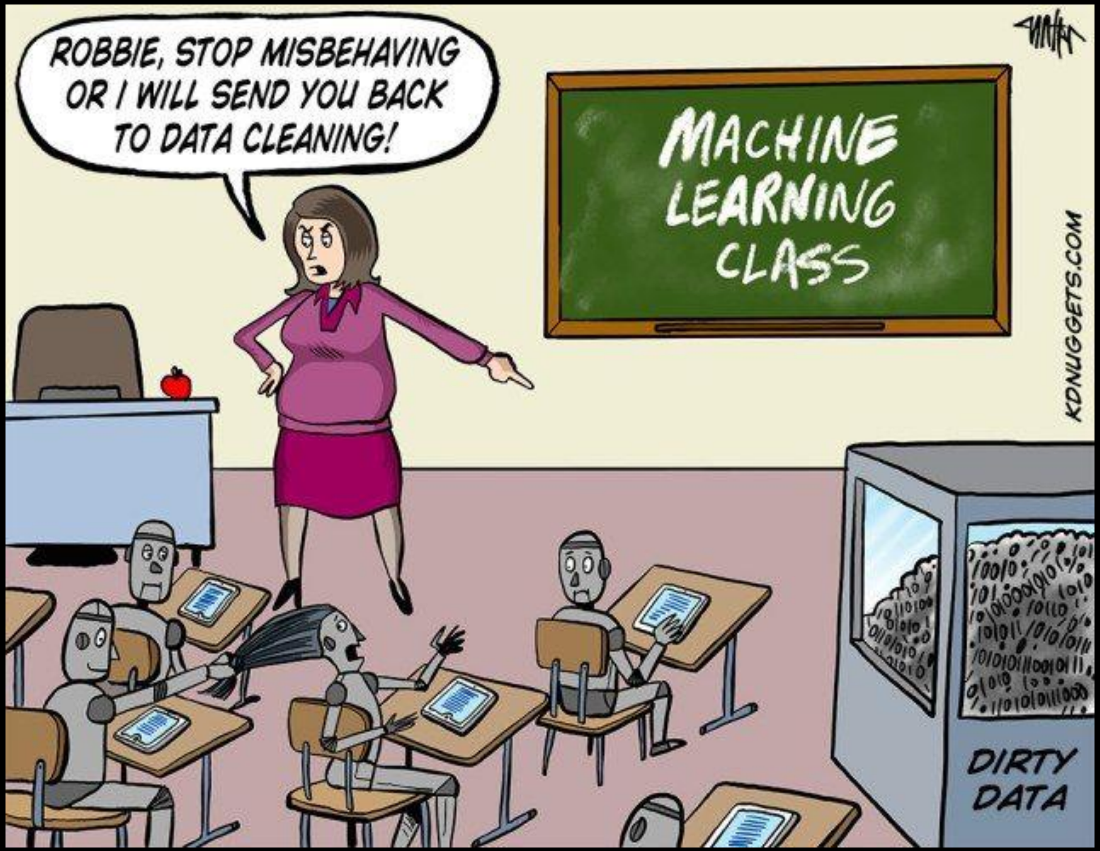


INFO 251: Applied Machine Learning

Missing and Imbalanced Data

Announcements

- Problem set 5 extended 1 week to April 4rd (midnight)

Key Concepts: Last Lecture

- “Deep” Learning
- Autoencoders – fully-connected and sparse
- Convolutions
- Pooling
- ReLU
- Convolutional Neural Networks
- Recurrent Neural Networks and LSTM’s

Course Outline

- Causal Inference and Research Design
  - Experimental methods
  - Non-experiment methods
- Machine Learning
  - Design of Machine Learning Experiments
  - Linear Models and Gradient Descent
  - Non-linear models
  - Neural models
  - **Practicalities**
  - Fairness and Bias
  - Unsupervised Learning
- Special topics

Key concepts: Today's lecture

- Stratified randomization
- Upsampling and downsampling
- SMOTE and AdaSyn
- Reweighting
- Algorithm-level adjustments for imbalance
- Problems with data missingness
- Selective labels
- Model-free imputation (e.g., zero-coding, mean imputation)
- Model-based imputation (e.g., hot deck)

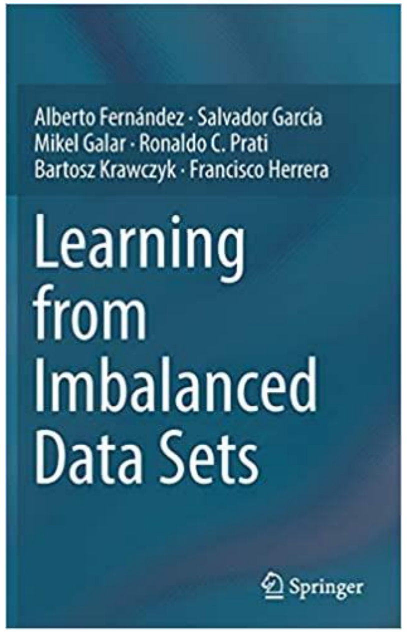
Outline

- **Imbalanced data: 4 ideas**
  - Simple things
  - Resampling
  - Weighting
  - Algorithm-level adjustments
- Missing data
  - Dropping
  - Imputing
  - “Selective labels”

Imbalanced data

- In many real-world instances, rate of true positives and true negatives are far from even
  - Credit card fraud
  - Disease diagnosis
  - Product adoption and churn
  - Terrorist threats

- *How to learn from imbalanced data?*



Imbalanced data

- Note that many learning algorithms will fail if used "out of the box" on imbalanced data
  - Most algorithms minimize *error*
  - A fast path to error minimization is predicting the majority class
    - Daume: *If a teacher told you to study for an exam with 1000 True/False questions and you knew that only one question had a correct answer of True, how much would you study?*

Imbalanced data

- Idea 1: Simple things
  - Don't rely on accuracy as a performance metric
  - Instead, use
    - Confusion matrices
    - ROC curves
    - Cohen's Kappa (normalizes accuracy by imbalance)

$$k = \frac{p_o - p_e}{1 - p_e}$$

- p_o = actual accuracy
- p_e = chance accuracy

    - F-scores, precision, recall, etc.

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Imbalanced data

- Idea 1: Simple things
  - Don't rely on accuracy as a performance metric
  - **Use appropriate baselines**

	Accuracy	Recall	Precision	F	AUC	% Answered Yes
Panel A: Assets and Housing						
Owns a radio	0.976	1.000	0.976	0.988	0.899	0.973
Owns a bicycle	0.676	0.552	0.678	0.609	0.722	0.456
Household has electricity	0.819	0.533	0.761	0.627	0.828	0.285
Owns a television	0.855	0.497	0.738	0.594	0.814	0.214
Has indoor plumbing	0.887	0.250	0.842	0.386	0.843	0.142
Owns a motorcycle/scooter	0.899	0.011	1.000	0.022	0.772	0.102
Owns a car/truck	0.945	0.213	0.867	0.342	0.849	0.068
Owns a refrigerator	0.954	0.180	1.000	0.305	0.878	0.055
Has landline telephone	0.992	0.125	1.000	0.222	0.562	0.009
Panel B: Social Welfare Indicators						
Hospital bills in last 12 months	0.633	0.890	0.633	0.740	0.653	0.587
Very ill in last 12 months	0.686	0.188	0.550	0.280	0.671	0.325
Death in family in last 12 months	0.665	0.183	0.632	0.284	0.619	0.363
Flood or drought in last 12 months	0.788	0.086	0.607	0.151	0.706	0.219
Fired in last 12 months	0.901	0.022	1.000	0.043	0.731	0.101

Table 2: Model performance at predicting responses from survey respondents based on call records data

10

Imbalanced data

- Idea 1: Simple things
  - Don't rely on accuracy as a performance metric
  - Use appropriate baselines
  - **Stratified randomization**
    - When randomizing into test-train, separately randomize minority class observations and majority class observations – ensures even proportions in test/train
    - Same goes for cross-validation, when randomly assigning observations to folds

11

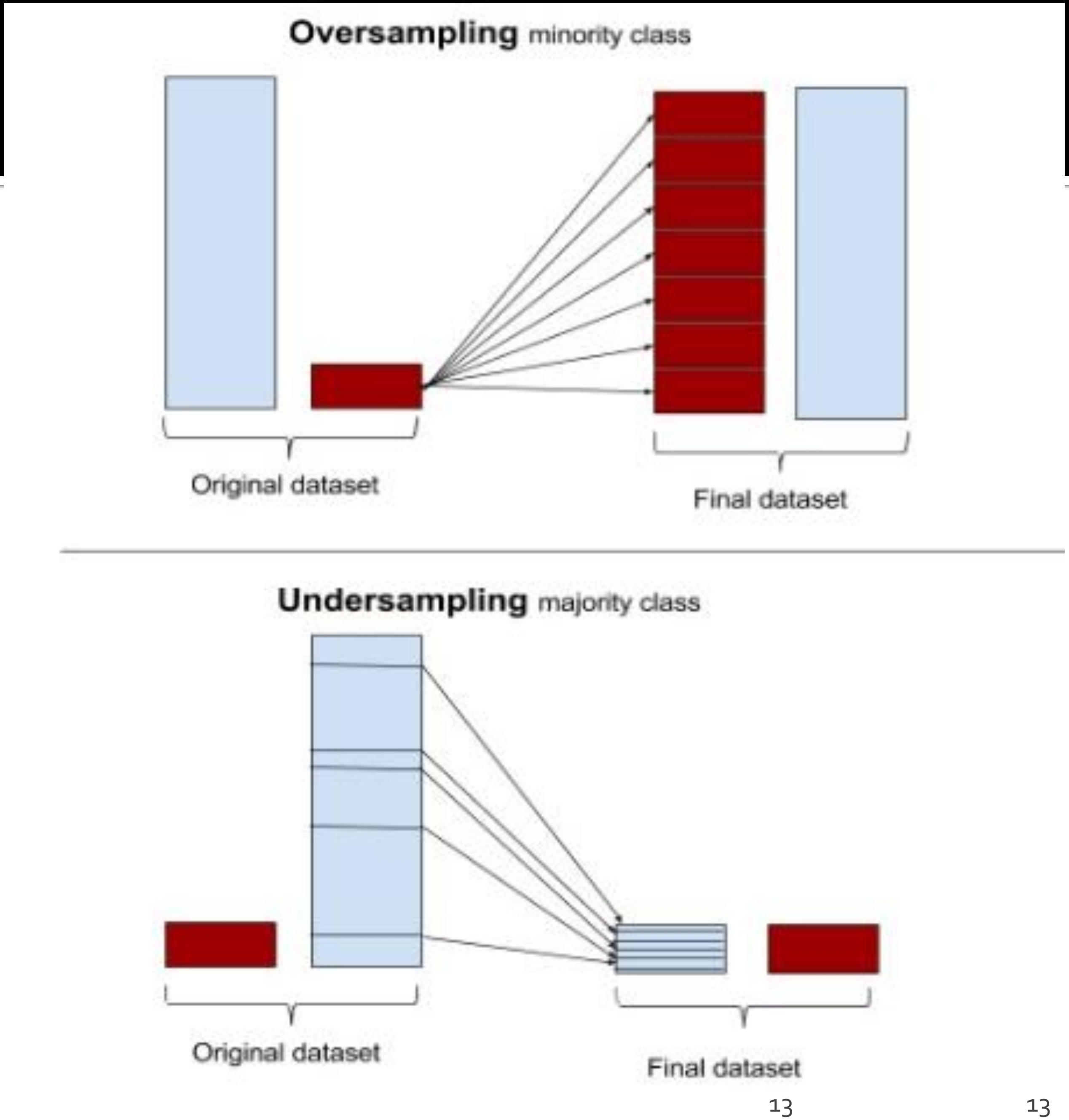
Imbalanced data

- Idea 1: Simple things
  - Don't rely on accuracy as a performance metric
  - Use appropriate baselines
  - Stratified randomization
  - **Adjust classification threshold**
    - E.g. Provost & Fawcett (1997): based on true class distributions and cost of (mis-)classifications
  - **Gather more data!**

12

Imbalanced data

- Idea 2: **Resampling**
  - Oversampling
  - Subsampling
  - Mixtures



Imbalanced data

- Idea 2: **Resampling**
  - Sub-sampling, or "**Down-sampling**"
    - Idea: Include all instances of minority class; include each instance of majority class with probability $1/\alpha$
    - This involves throwing out some data
    - Typically, set α = ratio of majority/minority class, to achieve 50-50 split of training data

Imbalanced data

- Idea 2: **Resampling**
  - Oversampling, or "**Up-sampling**"
    - Idea: Include all instances of majority class, include each instance of minority class α times
    - All data is used, some data is used several times

Imbalanced data

- Idea 2: **Resampling – Other options**
  - SMOTE: Synthetic Minority Over-sampling Technique
    - Creates artificial data based on the feature space similarities between existing minority examples.
  - ADASYN: ADAptive SYNthetic Sampling Approach
    - Creates more instances near mis-classified (minority) instances

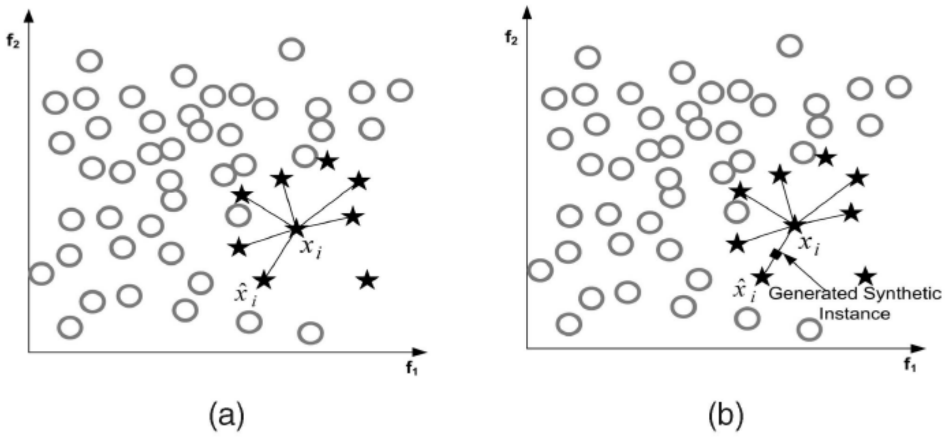


Fig. 3. (a) Example of the K-nearest neighbors for the x_i example under consideration ($K = 6$). (b) Data creation based on euclidian distance.

Imbalanced data

- Idea 2: **Resampling**
  - Other considerations
    - In both cases, *error rate* is (α times) higher than it will be on unbalanced classifier -- but other measures of performance often improve
    - Resampling typically done *after* train/test split
  - Comparing up- and down-sampling
    - Up-sampling often produces lower error (it sees more variance in training data!)
    - Computational efficiency is main reason for down-sampling
    - You can try both!

Imbalanced data

- Idea 3: **Weighting**
  - Instead of explicitly up- or down-sampling training data, associate a weight with each observation
  - For many algorithms this is straightforward and computationally efficient
    - k-Nearest Neighbors
    - Decision Trees / Random Forests (remember Adaboost?)
    - Regression
  - When possible, this is often the preferred approach

Imbalanced data

- Idea 4: **Algorithm-level adjustments**
  - Modify learning algorithm to reduce bias towards majority class. For example:
  - "Cost-sensitive" approaches modify the learner to vary penalty for different classes of examples, different (mis-)classifications, etc.
  - Also, many hybrid methods combine algorithm-level and data-level adjustments

Imbalanced data

- Other ideas
  - He, H., Garcia, E.A., 2009. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284.
  - Sun, Y., Wong, A.K.C., Kamel, M.S., 2009. Classification of imbalanced data: a review. Int. J. Patt. Recogn. Artif. Intell. 23, 687–719.

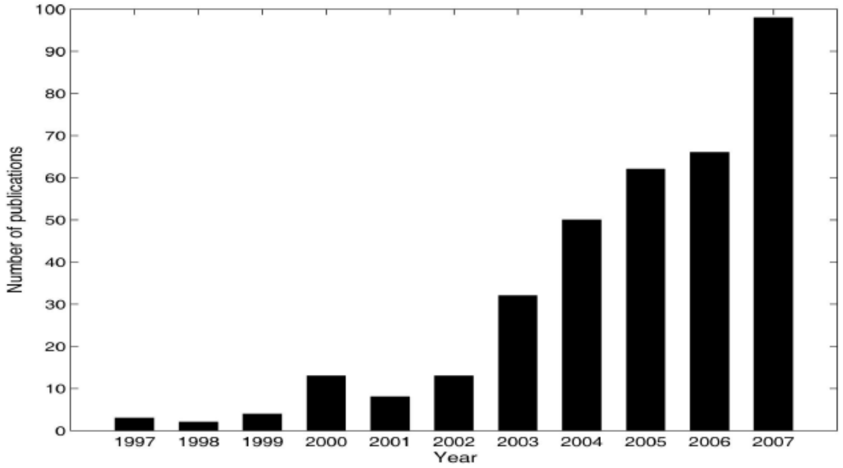


Fig. 1. Number of publications on imbalanced learning.

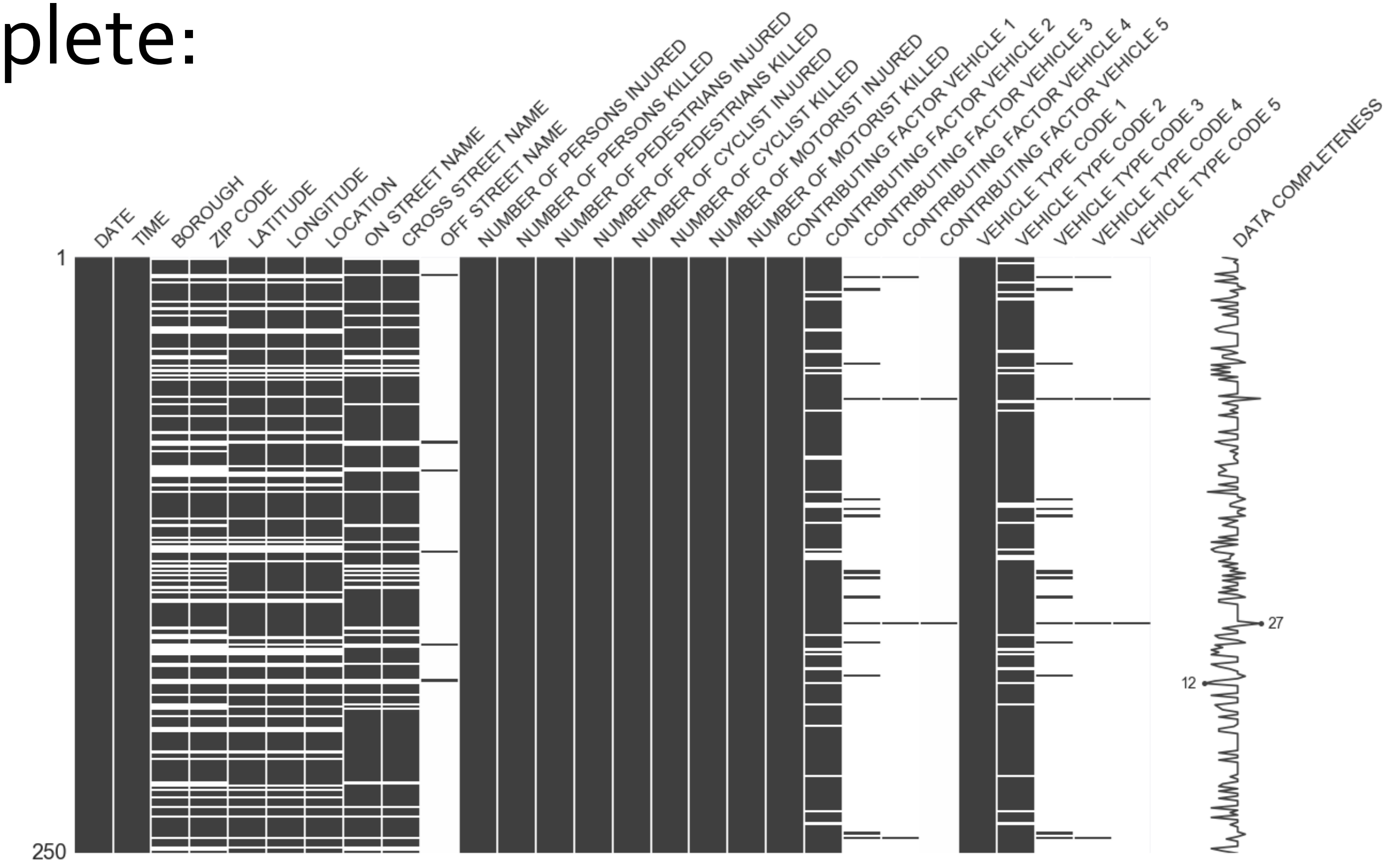
Outline

- Imbalanced data: 4 ideas
  - Simple things
  - Resampling
  - Weighting
  - Algorithm-level adjustments
- **Missing data**
  - Dropping
  - Imputing
  - "Selective labels"
- Multi-class classification

Missing data

- A common issue in applied ML settings is that a data matrix is frequently not complete:

```
import missingno as msno
msno.matrix(myData)
```



Missing data

- What to do?
  - Drop features
    - Great, if you can afford to do so!
  - Drop observations
    - Great, if you can afford to do so!
- In both cases, biggest concern is when data missingness is non-random
- Important: Many ML packages (e.g., sklearn) automatically drop observations and/or features with missing data!
  - Make sure to check!

Systematic attrition

- Systematic missingness: Example 1
  - We are interested in the effect of education on wages. So we estimate:

$$wages_i = \alpha + \beta * education_i + error_i$$

  - But wage data is systematically missing for certain people (e.g., the unemployed). What happens when we drop observations for people with missing wage data?

Systematic missingness

- Systematic missingness: Example 2
  - We are interested in predicting who will default on a loan. We drop all features where >50% of data are missing and train a predictive model:

$$default_i = f(X_i) + error_i$$

  - Several features get dropped: `last_loan, last_loan_value, last_loan_APR, last_loan_repaid`
  - What might go wrong here?

Missing data

- What to do?
  - Drop features
    - Great, if you can afford to do so!
  - Drop observations
    - Great, if you can afford to do so!
  - Create new variables
    - E.g., feature_K_is_present, and feature_K_is_present * feature_K
  - Impute, replace

Imputation

- Model-free approaches
  - Draw a value from distribution of X
  - Zero-coding, filling with marker (-9999)
  - Mean imputation
  - Interpolation, "carry-forward" (for panel data)
- Model-based approaches (done with training data!)
  - Regression modeling
  - Matching ("hot deck imputation")
  - k-NN imputation
  - (any other supervised algorithm can be used too!)

Selective labels

- One final note: The "selective labels problem"
  - We are interested in predicting who will default on a loan:

$$default_i = f(X_i) + error_i$$

  - We only can train on the population of people who have received loans in the past. Is this the same population for whom we want to make predictions?
  - Common settings
    - Bail decisions, recidivism
    - Hiring decisions
    - Admissions decisions
    - Cf. Lakkaraju et al. (2017 KDD)

Further reading

1. Introduction to KDD and Data Science
2. Foundations on Imbalanced Classification
3. Performance Measures
4. Cost-Sensitive Learning
5. Data Level Preprocessing Methods
6. Algorithm-Level Approaches
7. Ensemble Learning
8. Imbalanced Classification with Multiple Classes
9. Dimensionality Reduction for Imbalanced Learning
10. Data Intrinsic Characteristics
11. Learning from Imbalanced Data Streams
12. Non-classical Imbalanced Classification Problems
13. Imbalanced Classification for Big Data
14. Software and Libraries for Imbalanced Classification