Learning and Imbalanced Data

January 28, 2019

David Rimshnick

CORNELL TECH

What is data imbalance?

- Unequal distribution of data towards a certain characteristic
  - Target variable
    - Classification: Certain classes have much higher % of samples
      - E.g. Very rare disease, 99.9% of test results could be negative
    - Regression: Certain ranges of results much more prevalent
      - E.g. Almost all outputs are 0 or close to it, very few non-zero
  - Action variable
    - One of inputs (e.g. action) has very low variance in sample
    - Difficult for model to learn impact of changing that variable
    - Will revisit when we discuss reinforcement learning

Why is imbalance bad?

- Discussion

What is wrong with data imbalance

- Rare Disease Example: Classifier can get 99.9% accuracy by just assuming all negative!
  - This is also why accuracy is not the best metric
  - Loss function may need to be modified
  - Need to consider false-negative rate as well as true-positive, etc
  - *confusion matrix*, AuROC, etc – to be discussed again in later lectures
- Sample may not mimic population
  - 90% of sample is *A*, but only 50% of population is
- Overfitting - Model may 'memorize' defining characteristics of minority class instead of learning underlying pattern

How do we deal with data imbalance

- Alter the sample
  - Three primary methods:
    - Oversampling: For under-represented class or part of distribution, duplicate observations until dataset is balanced
    - Undersampling: For over-represented class or part of distribution, remove observations until dataset is balanced
    - Synthetic Data Creation
- Alter the cost function

Oversampling

- “Random Oversampling”
  - Randomly duplicate records from minority class(es) with replacement until dataset is balanced
  - Downside:

Oversampling

- "Random Oversampling"
  - Randomly duplicate records from minority class(es) with replacement until dataset is balanced
  - Downside: Overfitting
    - Model may 'memorize' idiosyncratic characteristics of overbalanced records as opposed to learning scalable pattern

Undersampling

- “Random Undersampling”
  - Randomly delete records from majority class(es) until dataset is balanced
  - Downside:

Undersampling

- "Random Undersampling"
  - Randomly delete records from majority class(es) until dataset is balanced
  - Downside: Loss of data!

‘Informed’ Undersampling

- Several methods exist (see paper for reference)
- Example: Edited Nearest Neighbor Rule (ENN)
  - Remove instance of majority class whose prediction made by KNN method is different than the majority class
    - Intuition: Remove “confusing” examples of majority class, make decision surface more smooth
    - Algorithm:
      1. Obtain the k nearest neighbors of x_i, $x_i \in N$
      2. x_i will be removed if the number of neighbors from another class is predominant
      3. The process will be repeated for every majority instance of the subset N

Synthetic Data Creation

- Instead of just resampling existing values to oversample, create artificial or synthetic data
- One of best known techniques: SMOTE (Synthetic Minority Over-sampling Technique)
  - Algorithm:
    - For each x_i from a minority set, choose n nearest neighbors
    - Select randomly one instance k from nearest neighbors
    - Create a new instance with features as a convex combination (with some parameter p) of the features of the original instance and the nearest neighbor

Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

Visualization of SMOTE algorithm

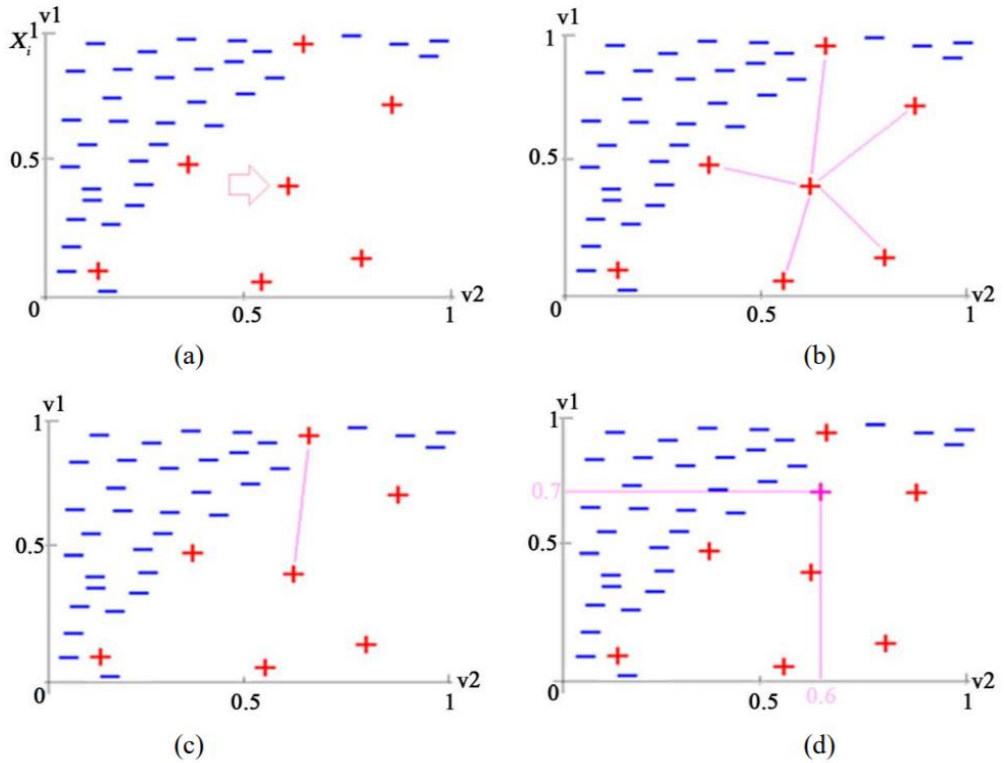

Figure 1. SMOTE process for $k = 5$. (a) An imbalanced dataset, with negative (−) and positive (+) instances. An instance x_i is selected; (b) the $k = 5$ nearest instances (neighbors) of x_i are selected; (c) one of the $k = 5$ neighbors $\hat{x}_i$, is randomly selected; (d) a new synthetic instance is created with the random values of v1 and v2 between x_i and $\hat{x}_i$.

Image Source:

Beckmann, Marcelo, Nelson FF Ebecken, and Beatriz SL Pires de Lima. "A KNN undersampling approach for data balancing." Journal of Intelligent Learning Systems and Applications 7.04 (2015): 104.

Cost function alteration

- Idea: Assign greater cost to observations from minority class
  - E.g.: In the loss function, assign weight $w_i = \frac{1}{p_i * C}$ where p_i is the sample proportion of class i, and C is the number of classes
  - Downside is that you have to edit algorithm, i.e. no longer black-box
- More general framework: Assign greater weight to observations that are mishandled by model
  - What is this technique when done iteratively?

Cost function alteration

- Idea: Assign greater cost to observations from minority class
  - E.g.: In the loss function, assign weight $w_i = \frac{1}{p_i * C}$ where p_i is the sample proportion of class i, and C is the number of classes
  - Downside is that you have to edit algorithm, i.e. no longer black-box
- More general framework: Assign greater weight to observations that are mishandled by model
  - What is this technique when done iteratively? Boosting!

Attribution

- This lecture is partially based on the following paper: H. He and E. A. Garcia, “Learning from Imbalanced Data,” IEEE Trans. Knowledge and Data Engineering, vol. 21, issue 9, pp. 1263-1284, 2009