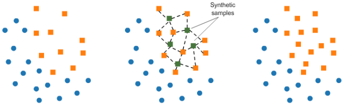
Advanced Machine Learning

Imbalanced Learning: Sampling Methods

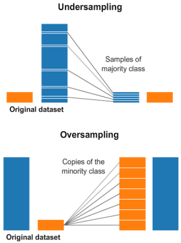


Learning goals

- Know the idea of sampling methods for coping with imbalanced data
- Understand the different undersampling techniques
- Understand the state-of-art oversampling technique SMOTE

SAMPLING METHODS: OVERVIEW

- Another popular way to deal with imbalanced data sets is to pre-process the data such that using standard classifiers (focusing on accuracy) is fine.
- The typical pre-processing approach is to use a *sampling technique*: one tries to manipulate the distribution of the training examples (make it more balanced) in order to improve the performance of the classifier on the minority classes.
- The advantage is that these techniques are independent of the underlying classifier making it a very flexible and general approach to deal with imbalanced data sets.
- Sampling techniques can be distinguished into three groups:
  - Undersampling — Eliminating/Removing instances of the majority class(es) in the original data set.
  - Oversampling — Adding/Creating new instances of the minority class(es) to the original data set.
  - Hybrid approaches — Combining undersampling and oversampling.



RANDOM UNDERSAMPLING/OVERSAMPLING

- A very common and oftentimes quite effective way of sampling is by doing simply random undersampling or random oversampling.

  - Random oversampling (ROS): Expand the minority.
  - Replicate uniformly at random instances from the minority class(es) until a desired imbalance ratio is reached.
  - Prone to overfitting due to multiple tied instances!

  - Random undersampling (RUS): Shrink the majority.
  - Eliminate uniformly at random instances from the majority class(es) until a desired imbalance ratio is reached.
  - This might remove informative data points and destroy the important concepts in the data!

- Better: Introduce heuristics in the removal process (RUS) and do not create exact copies (ROS).

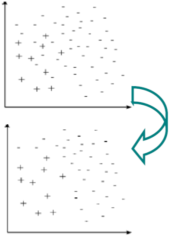
UNDERSAMPLING: TOMEK LINKS

- One idea to make undersampling more meaningful is to remove only noisy borderline examples of the majority class(es).
- Tomek link: Let $E^{(i)} = (\mathbf{x}^{(i)}, y^{(i)})$ and $E^{(j)} = (\mathbf{x}^{(j)}, y^{(j)})$ be two data points in $\mathcal{D}$ with $y^{(i)} \neq y^{(j)}$. A pair $(E^{(i)}, E^{(j)})$ is called *Tomek link* iff there is no other data point $E^{(k)} = (\mathbf{x}^{(k)}, y^{(k)})$ such that

  $$d(\mathbf{x}^{(i)}, \mathbf{x}^{(k)}) < d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$
  $$\text{or } d(\mathbf{x}^{(j)}, \mathbf{x}^{(k)}) < d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \text{ holds,}$$

  where d is some distance on $\mathcal{X}$.

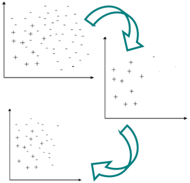
- Since $E^{(i)}$ and $E^{(j)}$ have different y's they correspond to a bordeline case.
- By removing each data pair in a Tomek link, where the y belongs to a majority class, we shrink the majority class(es) in the data set.
- Note that we do not sample here, but this approach can be combined with RUS.



Francisco Herrera (2013), Imbalanced Classification: Common Approaches and Open Problems (URL).

UNDERSAMPLING: CNN

- Another idea for shrinking the majority class is to remove examples (instances) from the majority class(es) which are far away from the decision boundary. This could be realized by constructing a consistent subset $\tilde{\mathcal{D}}$ of $\mathcal{D}$ in terms of the 1-NN classifier.
- A subset $\tilde{\mathcal{D}}$ of $\mathcal{D}$ is called consistent if using a 1-NN classifier on $\tilde{\mathcal{D}}$ classifies each instance in $\mathcal{D}$ correctly.
- The Condensed Nearest Neighbor (CNN) methods creates a consistent data subset by doing the following:
    1. Initialize $\tilde{\mathcal{D}}$ by selecting all minority class instances and by picking one majority class instance at random.
    2. Classify each instance in $\mathcal{D}$ with the 1-NN classifier based on $\tilde{\mathcal{D}}$.
    3. Move all misclassified instances from $\mathcal{D}$ to $\tilde{\mathcal{D}}$.



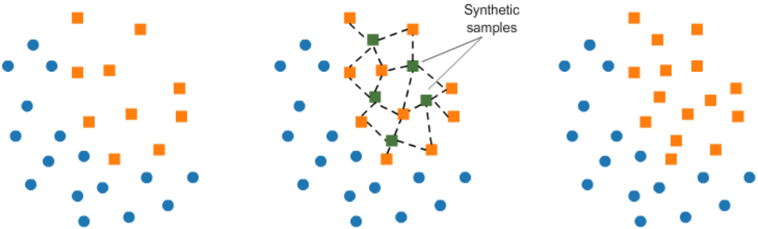
Francisco Herrera (2013), Imbalanced Classification: Common Approaches and Open Problems (URL).

UNDERSAMPLING: OTHER APPROACHES

- Neighborhood cleaning rule (NCL):
  1. Find the three nearest neighbors for each instance $(\mathbf{x}^{(i)}, y^{(i)})$ in $\mathcal{D}$.
  2. If $y^{(i)}$ belongs to the majority class *and* the three nearest neighbors classify it to be a minority class $\rightsquigarrow$ Remove $(\mathbf{x}^{(i)}, y^{(i)})$ from $\mathcal{D}$.
  3. If $y^{(i)}$ belongs to the minority class *and* the three nearest neighbors classify it to be a majority class $\rightsquigarrow$ Remove the three nearest neighbors from $\mathcal{D}$.
- One-sided selection (OSS): Tomek link + CNN
- CNN + Tomek link: Since finding Tomek links is computationally expensive, it would be more reasonable to first use CNN and then apply the reduction by removing the Tomek links on the reduced data set.
- Clustering approaches: Class Purity Maximization (CPM) and Undersampling based on Clustering (SBC).
- Near-Miss approaches: Techniques based on informed heuristics and 3-NN classification.
- See Fernández et al. (2018), Learning from imbalanced data sets.

OVERSAMPLING: SMOTE

- The Synthetic Minority Oversampling TEchnique (SMOTE) is an oversampling approach, where instead of replicating instances of the minority class one is creating new synthetic instances.
- Idea: Form new minority class examples by interpolating between several minority examples being close together.
- Note that the examples are created in the feature space $\mathcal{X}$ rather than in the data space $\mathcal{X} \times \mathcal{Y}$.
- Algorithm: For each minority class example
  - Find its *k* nearest minority neighbors.
  - Randomly select *j* of these neighbors.
  - Randomly generate new examples along the lines connecting the minority example and its *j* selected neighbors.



SMOTE: GENERATING NEW EXAMPLES

- Let $\mathbf{x}^{(i)}$ be the feature of the minority instance and let $\mathbf{x}^{(j)}$ be its nearest neighbor. The line segment connecting the two is given by

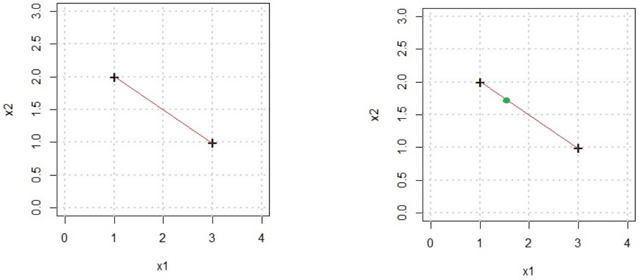
$$(1-\lambda)\mathbf{x}^{(i)} + \lambda\mathbf{x}^{(j)} = \mathbf{x}^{(i)} + \lambda(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})$$

  where $\lambda \in [0,1]$.

- Thus, by sampling randomly a $\lambda \in [0,1]$, say $\tilde{\lambda}$, we can create a new example on the connecting line via

$$\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} + \tilde{\lambda}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})$$

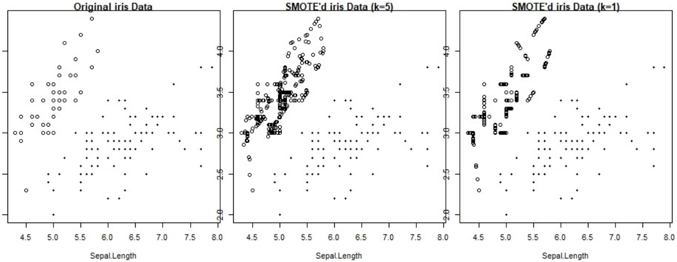
Example: Let $\mathbf{x}^{(i)} = (1,2)^\top$ and $\mathbf{x}^{(j)} = (3,1)^\top$. Assume we draw $\tilde{\lambda} \approx 0.25$.



SMOTE: EXAMPLE

- Let us consider the iris data set with $\mathcal{Y} = \{\texttt{setosa}, \texttt{versicolor}, \texttt{virginica}\}$ with 50 examples for each class.
- We can make the data set "imbalanced" by encoding each instance of one class as positive and the remaining instances of the two other classes as negative.

  NB: This is a common approach to create imbalanced data sets artificially.
- Running SMOTE with different *k*'s leads to the following:



We created now (minority) data points which make it now slightly harder to separate the two classes with a separating hyperplane.

SMOTE: DIS-/ADVANTAGES

- SMOTE's oversampling approach generalizes the decision region for the minority class instead of making it quite specific such as by random oversampling.
- SMOTE is, however, prone to overgeneralizing as it generalizes the minority area without paying attention to the majority class(es).
- Nevertheless, SMOTE still belongs to the state-of-art techniques to deal with imbalanced data sets via oversampling and is the basis for a plethora of other oversampling methods oftentimes tailored towards specific data situations: Borderline-SMOTE, LN-SMOTE, . . . (over 90 extensions!)
- A quite common approach is also to combine SMOTE with undersampling techniques such as Tomek link or NCL.