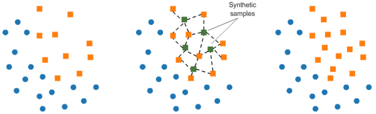
Applied Machine Learning

Imbalanced Data: Sampling Methods



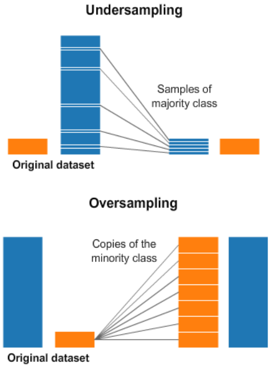
Learning goals

- Understand under- and oversampling strategies
- Apply SMOTE for synthetic minority class generation
- Compare different sampling approaches and their trade-offs

Under- and Oversampling

SAMPLING METHODS: OVERVIEW

- Balance training data distribution to perform better on minority classes.
- Independent of classifier $\rightsquigarrow$ very flexible and general.
- Three groups:
  - Undersampling — Removing instances of majority class(es).
  - Oversampling — Adding/Creating new instances of minority class(es). (Slower, but usually works better.)
  - Hybrid — Combining both methods.



RANDOM UNDERSAMPLING/OVERSAMPLING

- Random oversampling (ROS):
  - Randomly **replicate minority** instances.
  - Prone to overfitting due to multiple tied instances.
- Random undersampling (RUS):
  - Randomly **eliminate majority** instances.
  - Might remove informative instances and destroy important concepts in data.
- Better: Introduce heuristics in removal process (RUS) and do not create exact copies (ROS).

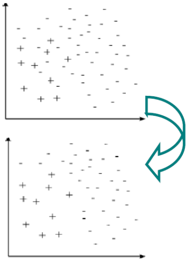
UNDERSAMPLING: TOMEK LINKS

- Remove "noisy borderline" examples (very close observations of different classes) of majority class(es).
- Let $E^{(i)} = (\mathbf{x}^{(i)}, y^{(i)})$ and $E^{(j)} = (\mathbf{x}^{(j)}, y^{(j)})$ be two data points in $\mathcal{D}$.

- 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 $\mathcal{X}$.
- $y^{(i)} \neq y^{(j)} \rightsquigarrow$ noisy borderline examples.
- Remove majority instance in each data pair in a Tomek link where $y^{(i)} \neq y^{(j)}$.



Franciso Herrera (2013), Imbalanced Classification: Common Approaches and Open Problems ▸ "Herrera" 2013.

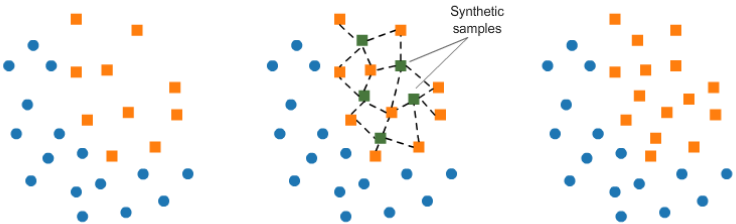
UNDERSAMPLING: OTHER APPROACHES

- Neighborhood cleaning rule (NCL):
  1. Find 3 nearest neighbors for each $(\mathbf{x}^{(i)}, y^{(i)})$ in $\mathcal{D}$.
  2. If $y^{(i)}$ is majority class *and* 3-NN classifies it as minority $\rightsquigarrow$ Remove $(\mathbf{x}^{(i)}, y^{(i)})$ from $\mathcal{D}$.
  3. If $y^{(i)}$ is minority class *and* 3-NN classifies it as majority $\rightsquigarrow$ Remove 3 nearest neighbors from $\mathcal{D}$.
- Condensed Nearest Neighbor (CNN): Construct a **minimally consistent** subset $\tilde{\mathcal{D}}$ of $\mathcal{D}$.
- One-sided selection (OSS): Tomek link + CNN
- CNN + Tomek link: to reduce computation of finding Tomek links $\rightsquigarrow$ first use CNN and then remove the Tomek links.
- Clustering approaches: Class Purity Maximization (CPM) and Undersampling based on Clustering (SBC).

SMOTE

OVERSAMPLING: SMOTE

- SMOTE creates **synthetic instances** of minority class.
- Interpolate between neighboring minority instances.
- Instances are created in $\mathcal{X}$ rather than in $\mathcal{X} \times \mathcal{Y}$.
- Algorithm: For each minority class instance:
  - Find its *k* nearest minority neighbors.
  - Randomly select one of these neighbors.
  - Randomly generate new instances along the lines connecting the minority example and its selected neighbor.



SMOTE: GENERATING NEW EXAMPLES

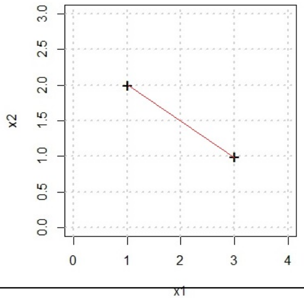
- Let $\mathbf{x}^{(i)}$ be the feature of the minority instance and let $\mathbf{x}^{(j)}$ be its nearest neighbor. The line connecting the two instances is

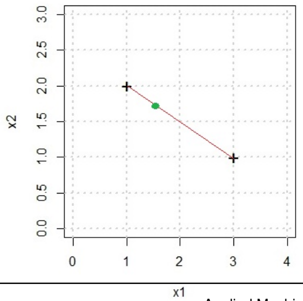
$$(1-\lambda)\mathbf{x}^{(i)} + \lambda\mathbf{x}^{(j)} = \mathbf{x}^{(i)} + \lambda(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})$$

  where $\lambda \in [0, 1]$.
- By sampling a $\lambda \in [0, 1]$, say $\tilde{\lambda}$, we create a new instance

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

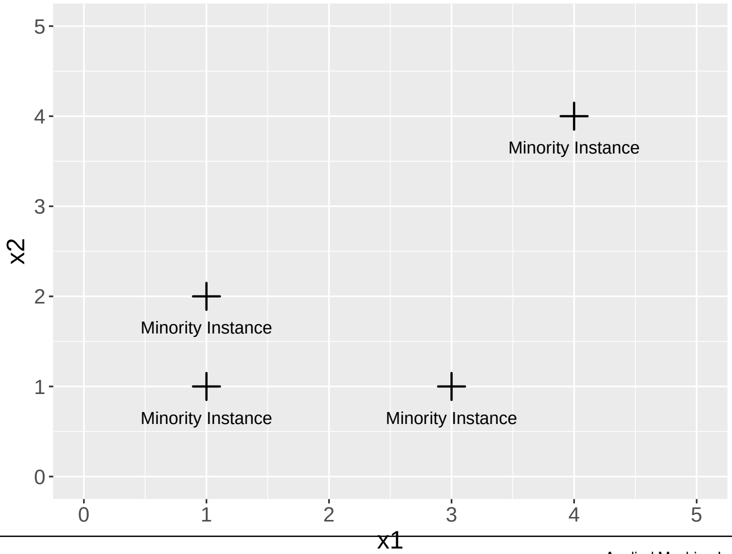
Example: Let $\mathbf{x}^{(i)} = (1, 2)^\top$ and $\mathbf{x}^{(j)} = (3, 1)^\top$. Assume $\tilde{\lambda} \approx 0.25$.





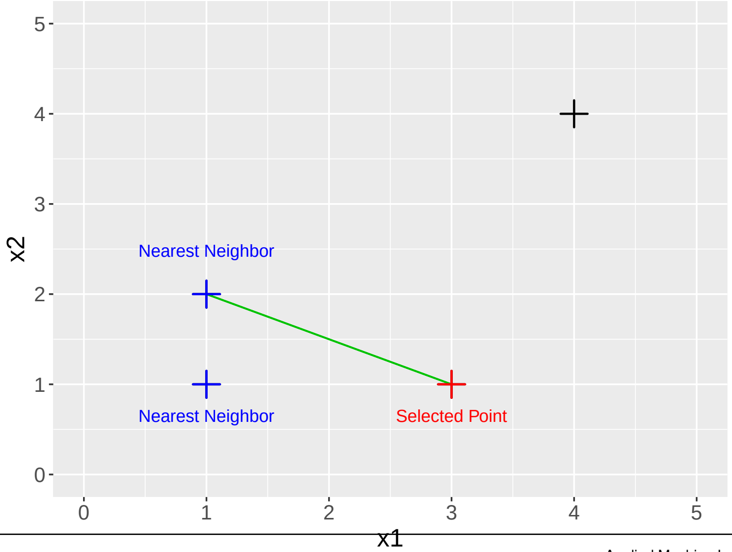
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



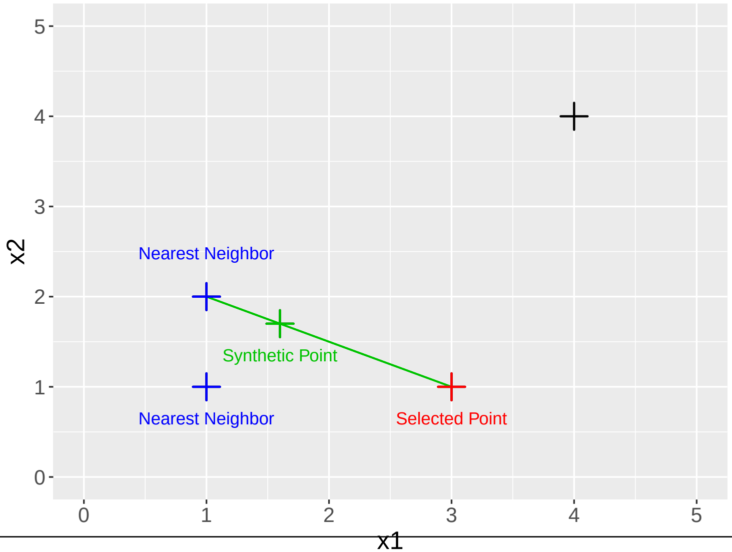
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



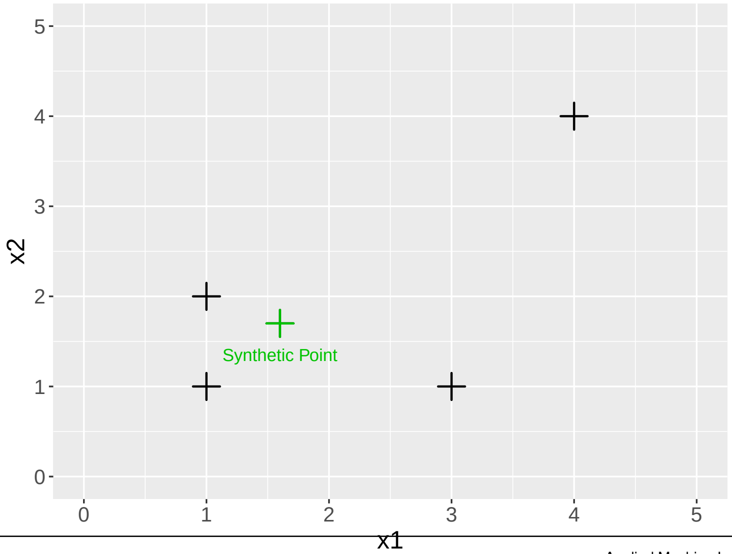
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



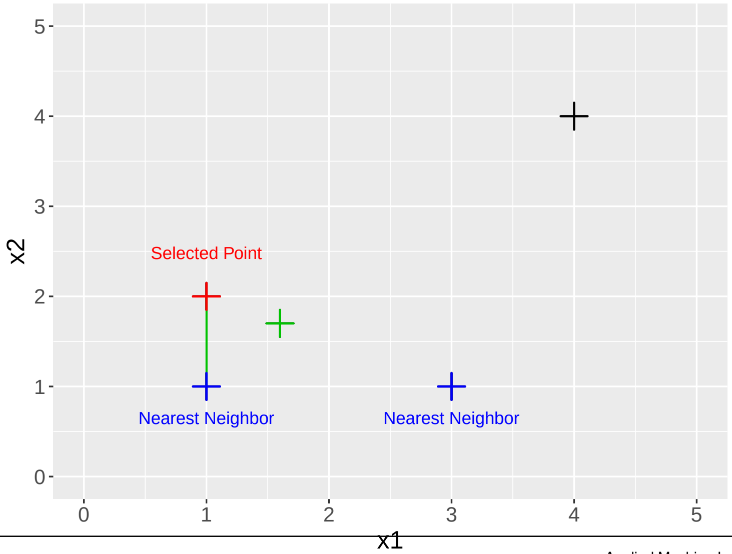
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



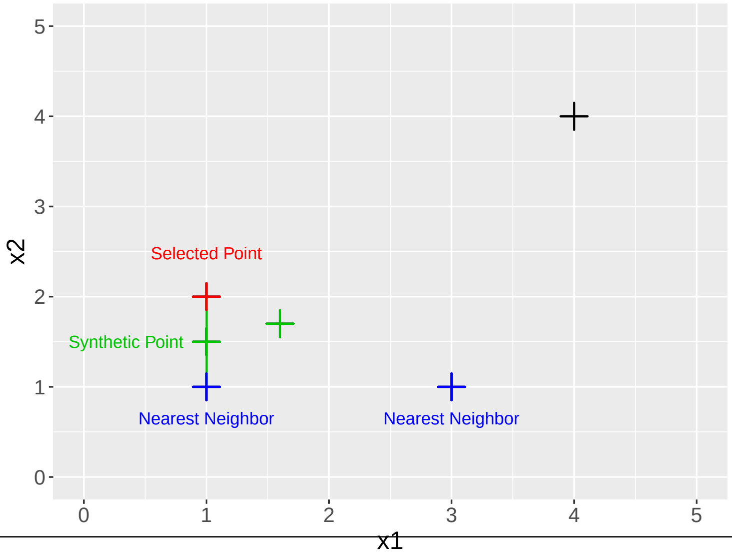
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



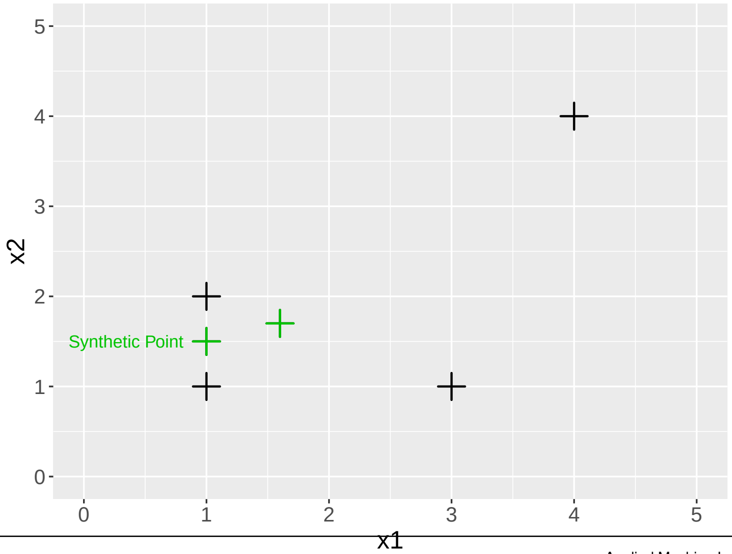
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



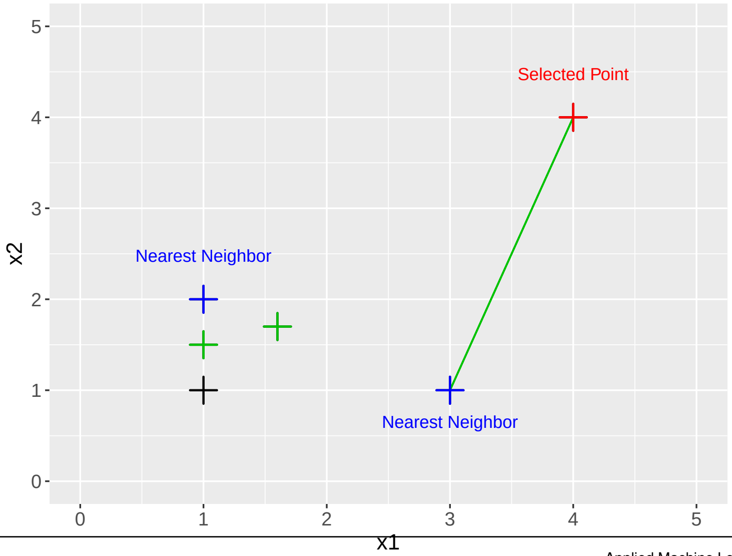
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



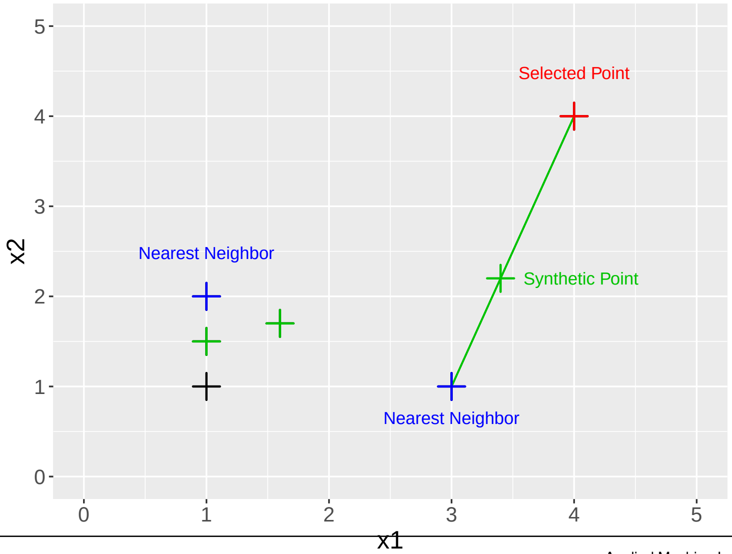
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



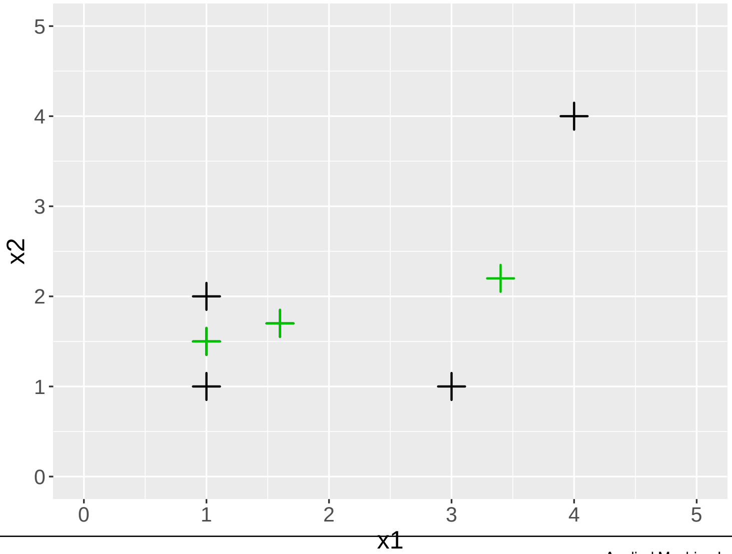
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



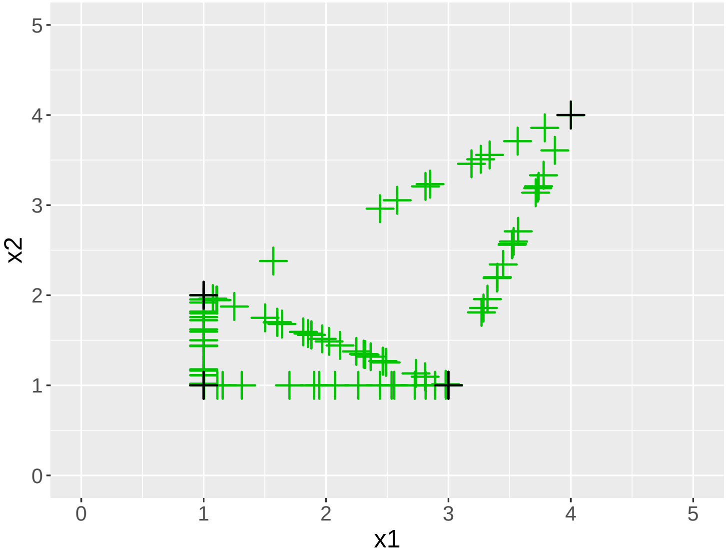
SMOTE: VISUALIZATION

For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



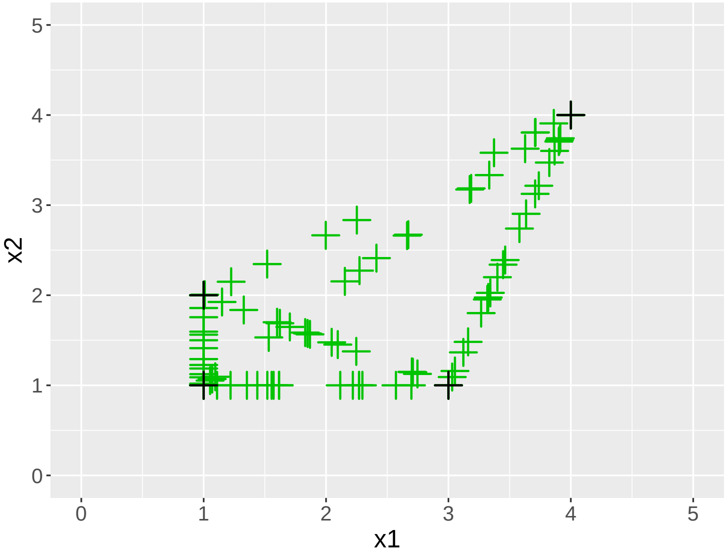
SMOTE: VISUALIZATION CONTINUED

After 100 iterations of SMOTE for $K = 2$ we get:



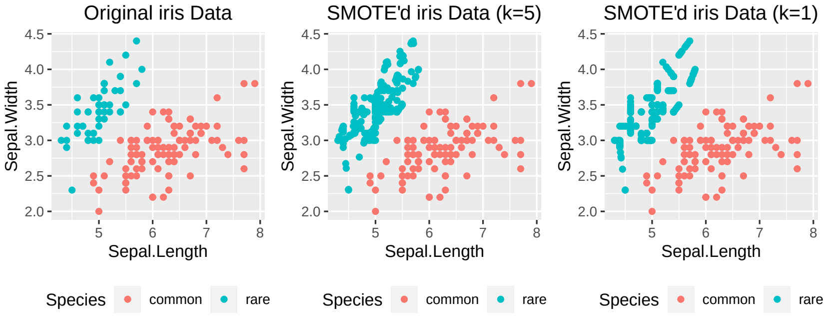
SMOTE: VISUALIZATION CONTINUED

After 100 iterations of SMOTE for $K = 3$ we get:



SMOTE: EXAMPLE

- Iris data set with 3 classes and 50 instances per class.
- Make the data set “imbalanced”:
  - relabel one class as positive
  - relabel two other classes as negative



SMOTE enriches minority class feature space.

SMOTE: DIS-/ADVANTAGES

- Generalize decision region for minority class instead of making it quite specific, such as by random oversampling.
- Well-performed among the oversampling techniques and is the basis for many oversampling methods: Borderline-SMOTE, LN-SMOTE, . . . (over 90 extensions!)
- Prone to overgeneralizing as it pays no attention to majority class.

COMPARISON OF SAMPLING TECHNIQUES

- Compare different sampling techniques on a binarized version of Optdigits dataset for optical recognition of handwritten digits.
- Use random forest with 100 trees, 5-fold cv, and F_1-Score.

Sampling technique	Class ratio	F1-Score
None	0.11	0.9239
Undersampling	0.68	0.9538
Oversampling	0.69	0.9538
SMOTE	0.79	0.9576

- Class ratios could be tuned (here done manually).
- Sampling techniques outperform base learner.
- SMOTE leads sampling techniques, although by a small margin.

Conclusion

WHEN TO COUNTERACT IMBALANCED DATA?

- Only counteract if your metric is impacted by imbalanced data
- How to counteract? Can you change to a metric that is not affected by imbalanced data?
- Check if treatment of imbalanced data has any adversarial effects
  - ▸ “Adversarial Effects of Imbalanced Data Treatment” 2024
- Try simple methods first, especially SMOTE is highly criticized
  - ▸ “Critical Analysis of SMOTE” 2022
- Use hyperparameter optimization to decide what method to use.
- Why not treat finding the trade-off between precision and recall as a multi-objective hyperparameter optimization problem?