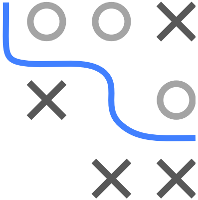
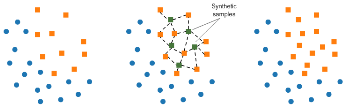
# Advanced Machine Learning

# Imbalanced Learning: Sampling Methods Part 2



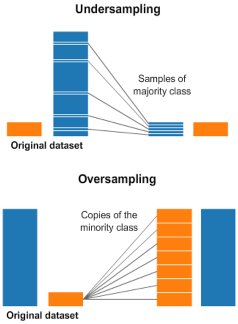



## Learning goals

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

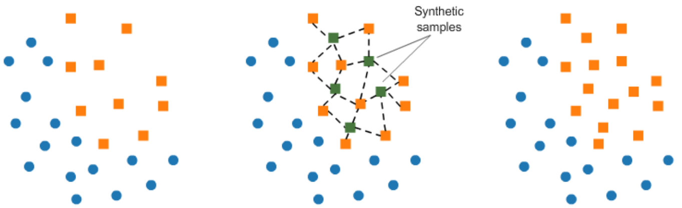
# SAMPLING METHODS: OVERVIEW

- Balance training data distribution to perform better on minority classes.
- Independent of classifier $\rightsquigarrow$ very flexible and general.
- Three groups:
  - Undersampling — Removing majority instances.
  - Oversampling — Adding/Creating new minority instances.
  - Oversampling is slower than undersampling but usually works better.
  - Hybrid approaches — Combining undersampling and oversampling.



# OVERSAMPLING: SMOTE

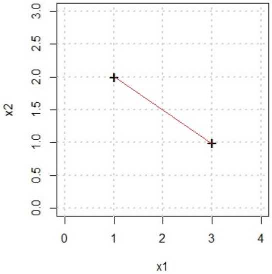
- SMOTE operates by creating **new synthetic instances** of minority class.
- Interpolate between neighboring minority instances.
- Instances are created in $\mathcal{X}$ rather than in $\mathcal{X} \times \mathcal{Y}$.
- Algorithm: For each minority class instance:
  - Find its *k* nearest minority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

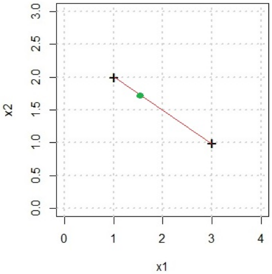


# SMOTE: GENERATING NEW EXAMPLES

- Let $\mathbf{x}^{(i)}$ be the feature of the minority instance and let $\mathbf{x}^{(j)}$ be its nearest neighbor. The line 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
$$\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} + \tilde{\lambda}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})$$

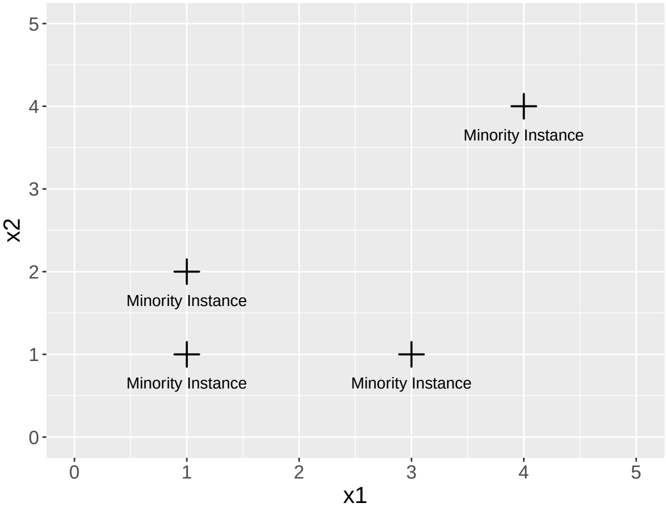
Example: Let $\mathbf{x}^{(i)} = (1, 2)^\top$ and $\mathbf{x}^{(j)} = (3, 1)^\top$. Assume $\tilde{\lambda} \approx 0.25$.

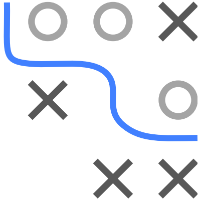


# SMOTE: VISUALIZATION

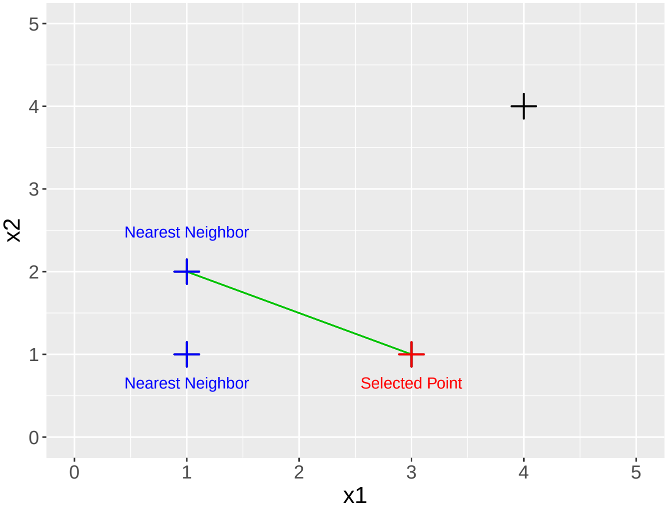
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.

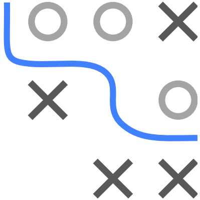


# SMOTE: VISUALIZATION

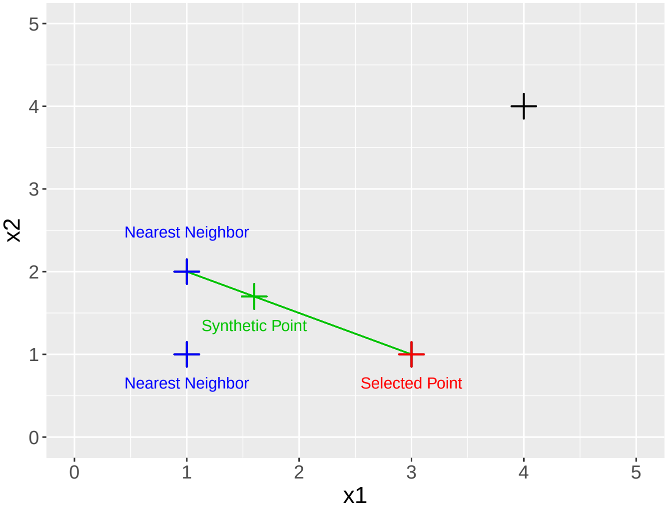
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.

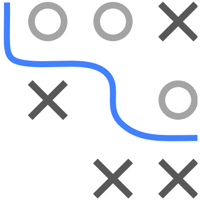


# SMOTE: VISUALIZATION

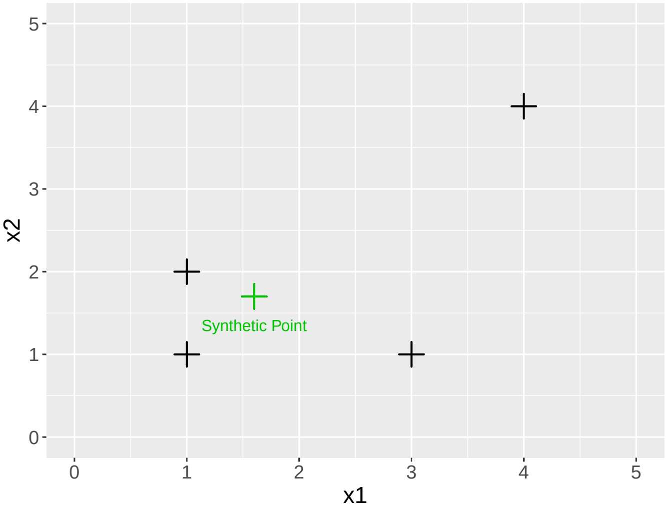
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.

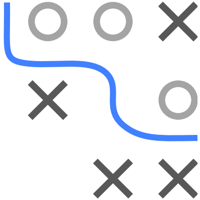


# SMOTE: VISUALIZATION

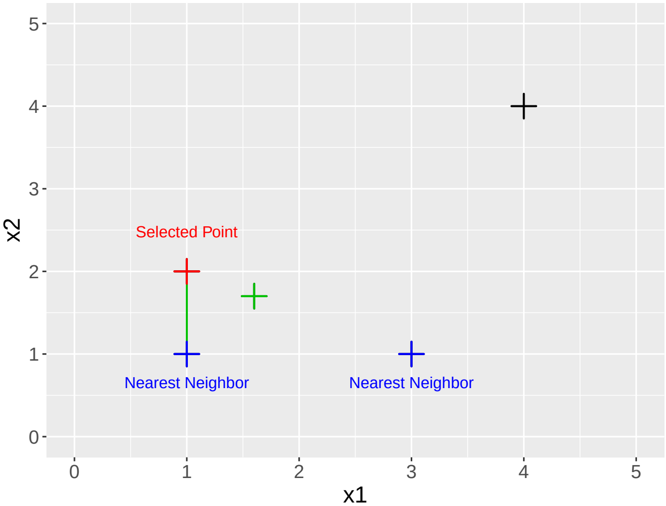
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.

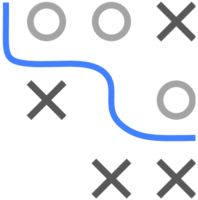


# SMOTE: VISUALIZATION

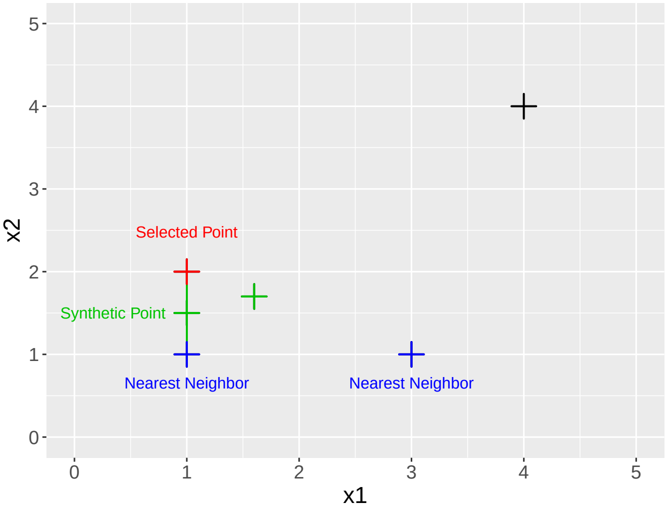
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.

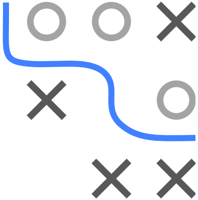


# SMOTE: VISUALIZATION

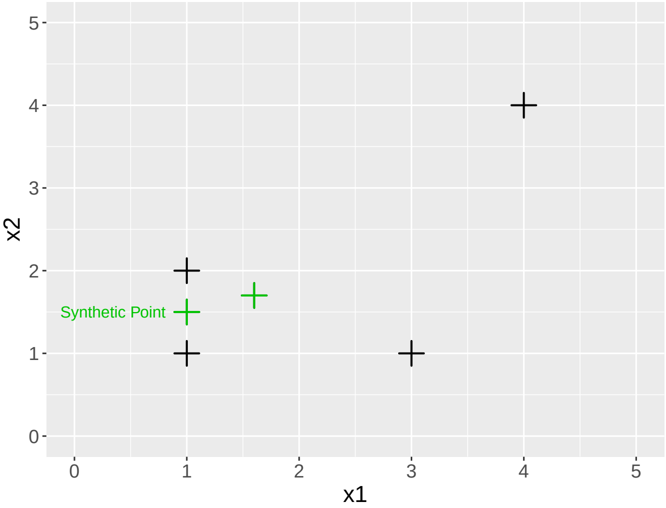
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



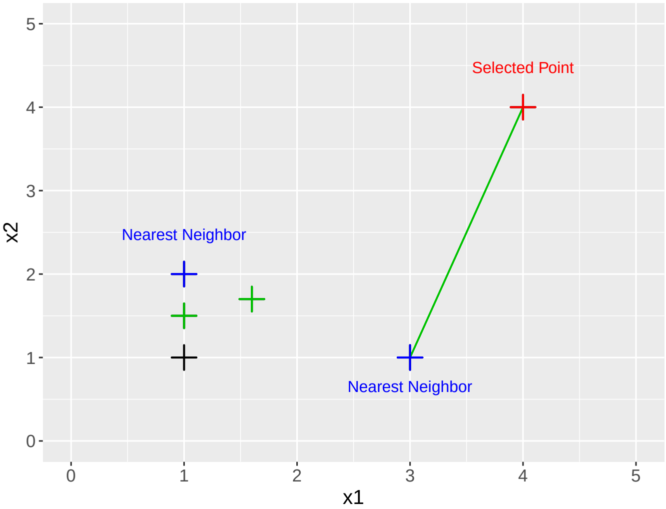
# SMOTE: VISUALIZATION

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



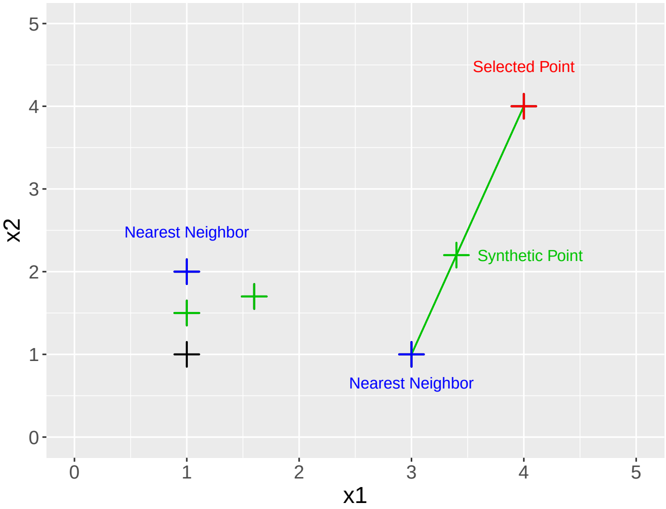
# SMOTE: VISUALIZATION

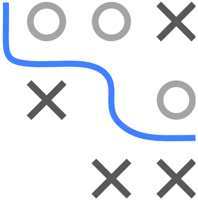
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



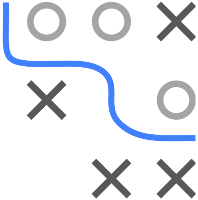
# SMOTE: VISUALIZATION

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

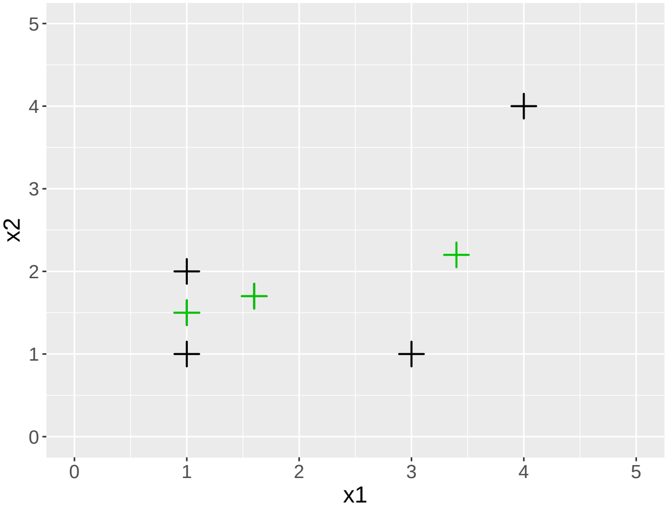


# SMOTE: VISUALIZATION

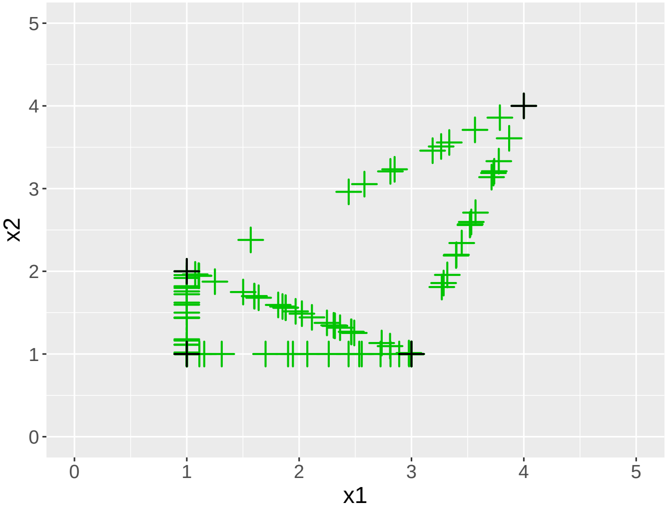


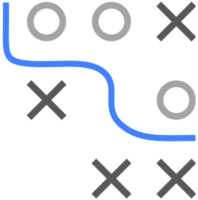
For an imbalanced data situation, take four instances of the minority class. Let $K = 2$ be the number of nearest neighbors.



# SMOTE: VISUALIZATION CONTINUED

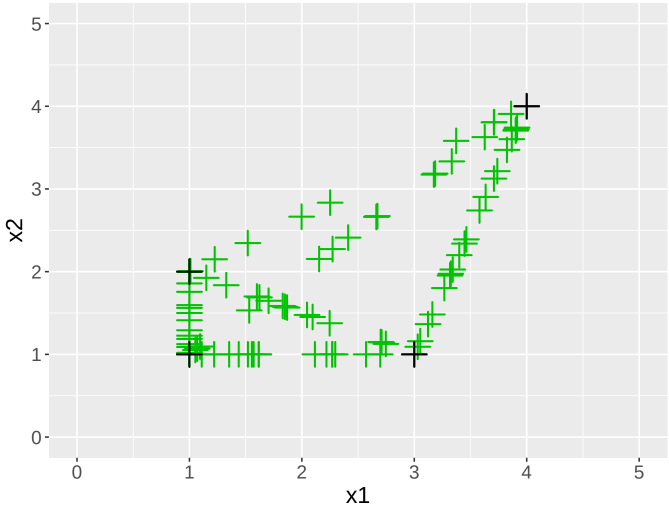
After 100 iterations of SMOTE for $K = 2$ we get:



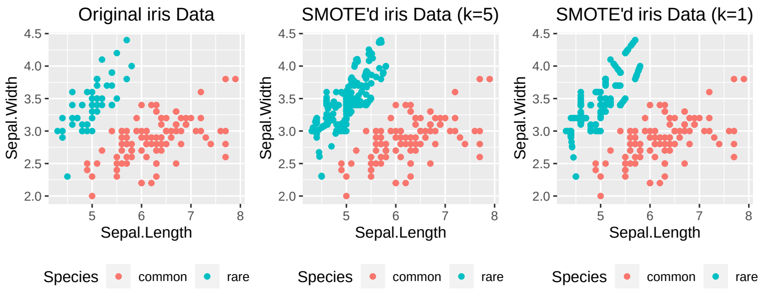
# SMOTE: VISUALIZATION CONTINUED

For 100 iterations of SMOTE with $K = 3$ and randomly selecting neighbors:



# SMOTE: EXAMPLE

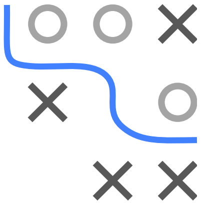
- Iris data set with $\mathcal{Y} = \{\texttt{setosa}, \texttt{versicolor}, \texttt{virginica}\}$, and 50 instances for each class.
- Make the data set “imbalanced”:
  - relabel one class as positive
  - relabel two other classes as negative




The new minority instances slightly increase the difficulty of separating the two classes with a hyperplane.

# 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 the majority class.



# COMPARISON OF SAMPLING TECHNIQUES

- Compare different sampling techniques on the Optdigits dataset for optical recognition of handwritten digits.
- Use random forest with 100 trees, 5-fold cv and employ $F_1$-Score. The pos./neg. class-ratios are 0.11, 0.68, 0.68 and 0.79:

| Learner | F1-Score |
|---|---:|
| Base RF | 0.9239 |
| Undersample RF | 0.9538 |
| Oversample RF | 0.9538 |
| SMOTE RF | 0.9576 |

- Sampling techniques outperform base learner.
- SMOTE leads sampling techniques, although by a small margin.