CS685: Data Mining
Lazy Learners and Class Imbalance

Arnab Bhattacharya
arnabb@cse.iitk.ac.in

Computer Science and Engineering,
Indian Institute of Technology, Kanpur
http://web.cse.iitk.ac.in/~cs685/

1st semester, 2021-22

Mon 1030-1200 (online)

Lazy Learners

- Till now, all classifiers were eager learners
- They built a model *eagerly* as soon as training data was available

Lazy Learners

- Till now, all classifiers were eager learners
- They built a model *eagerly* as soon as training data was available
- Lazy learners do *not* construct any model when training data comes
- It simply lets the training data be
    - It may do some indexing, etc.
- Only when a test object arrives, it uses the training data
- Also called instance-based classifiers as the work done is based on the test instance

Lazy Learners

- Till now, all classifiers were eager learners
- They built a model *eagerly* as soon as training data was available
- Lazy learners do *not* construct any model when training data comes
- It simply lets the training data be
    - It may do some indexing, etc.
- Only when a test object arrives, it uses the training data
- Also called instance-based classifiers as the work done is based on the test instance
- Examples
    - K-nearest-neighbor (kNN) classifier
    - Case-based reasoning (CBR) classifier

K-Nearest-Neighbor (KNN) Classifier

- When a test object comes, find its k nearest neighbors from the training set
  - If it looks like a duck, swims like a duck, quacks like a duck, then it probably is a duck
- *Majority* class of the k nearest neighbors is assigned its class

K-Nearest-Neighbor (KNN) Classifier

- When a test object comes, find its k nearest neighbors from the training set
  - If it looks like a duck, swims like a duck, quacks like a duck, then it probably is a duck
- *Majority* class of the k nearest neighbors is assigned its class
- Increment function in majority may be *weighted*
- Weights may be inverse of the distance from the test object

K-Nearest-Neighbor (KNN) Classifier

- When a test object comes, find its k nearest neighbors from the training set
    - If it looks like a duck, swims like a duck, quacks like a duck, then it probably is a duck
- *Majority* class of the k nearest neighbors is assigned its class
- Increment function in majority may be *weighted*
- Weights may be inverse of the distance from the test object
- What is size of k?

K-Nearest-Neighbor (KNN) Classifier

- When a test object comes, find its k nearest neighbors from the training set
  - If it looks like a duck, swims like a duck, quacks like a duck, then it probably is a duck
- *Majority* class of the k nearest neighbors is assigned its class
- Increment function in majority may be *weighted*
- Weights may be inverse of the distance from the test object
- What is size of k?
- If k is too small, susceptible to noise points
- If k is too large, neighborhood property is lost

K-Nearest-Neighbor (KNN) Classifier

- When a test object comes, find its k nearest neighbors from the training set
  - If it looks like a duck, swims like a duck, quacks like a duck, then it probably is a duck
- *Majority* class of the k nearest neighbors is assigned its class
- Increment function in majority may be *weighted*
- Weights may be inverse of the distance from the test object
- What is size of k?
- If k is too small, susceptible to noise points
- If k is too large, neighborhood property is lost
- k can be determined empirically as the one that gives the best result in the training set

K-Nearest-Neighbor (KNN) Classifier

- When a test object comes, find its k nearest neighbors from the training set
  - If it looks like a duck, swims like a duck, quacks like a duck, then it probably is a duck
- *Majority* class of the k nearest neighbors is assigned its class
- Increment function in majority may be *weighted*
- Weights may be inverse of the distance from the test object
- What is size of k?
- If k is too small, susceptible to noise points
- If k is too large, neighborhood property is lost
- k can be determined empirically as the one that gives the best result in the training set
- Finding k nearest neighbors is costly without any pre-processing or indexing

Case-Based Reasoning (CBR) Classifier

- When a test object comes, find its *exact match* within the training set
- Assign its class
- If no exact match is found, then unclassified

Case-Based Reasoning (CBR) Classifier

- When a test object comes, find its *exact match* within the training set
- Assign its class
- If no exact match is found, then unclassified
- Alternatively, find maximum number of attributes that match

Case-Based Reasoning (CBR) Classifier

- When a test object comes, find its *exact match* within the training set
- Assign its class
- If no exact match is found, then unclassified
- Alternatively, find maximum number of attributes that match
- Model each training instance as a graph
- The test object is also a graph
- Find the training instance that corresponds to the maximal common subgraph
- Assign its class

Case-Based Reasoning (CBR) Classifier

- When a test object comes, find its *exact match* within the training set
- Assign its class
- If no exact match is found, then unclassified
- Alternatively, find maximum number of attributes that match
- Model each training instance as a graph
- The test object is also a graph
- Find the training instance that corresponds to the maximal common subgraph
- Assign its class
- How to model as a graph depends on the application

Case-Based Reasoning (CBR) Classifier

- When a test object comes, find its *exact match* within the training set
- Assign its class
- If no exact match is found, then unclassified
- Alternatively, find maximum number of attributes that match
- Model each training instance as a graph
- The test object is also a graph
- Find the training instance that corresponds to the maximal common subgraph
- Assign its class
- How to model as a graph depends on the application
- Can be very costly without any pre-processing or indexing

Case-Based Reasoning (CBR) Classifier

- When a test object comes, find its *exact match* within the training set
- Assign its class
- If no exact match is found, then unclassified
- Alternatively, find maximum number of attributes that match
- Model each training instance as a graph
- The test object is also a graph
- Find the training instance that corresponds to the maximal common subgraph
- Assign its class
- How to model as a graph depends on the application
- Can be very costly without any pre-processing or indexing
- Can be considered as a special case of KNN classifier

Class Imbalance

- Class imbalance in training data is a practical problem
- Measures like accuracy become meaningless
    - If there is 90% of class-1 and 10% of class-2, then simply guessing class-1 gives 90% accuracy

Class Imbalance

- Class imbalance in training data is a practical problem
- Measures like accuracy become meaningless
  - If there is 90% of class-1 and 10% of class-2, then simply guessing class-1 gives 90% accuracy
- Under-sampling the more frequent class
  - Randomly selecting a subset
  - May underfit

Class Imbalance

- Class imbalance in training data is a practical problem
- Measures like accuracy become meaningless
  - If there is 90% of class-1 and 10% of class-2, then simply guessing class-1 gives 90% accuracy
- Under-sampling the more frequent class
  - Randomly selecting a subset
  - May underfit
- Over-sampling the less frequent class
  - Ends up choosing a point multiple times
  - May not help or may overfit

SMOTE

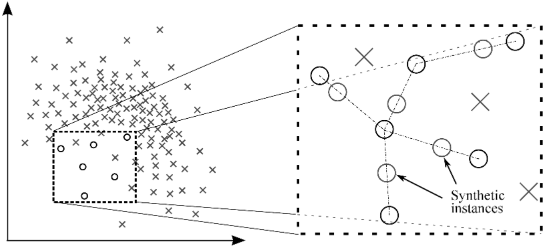
- SMOTE algorithm (Synthetic Minority Over-Sampling Technique)
- Generates synthetic examples in the less frequent class

SMOTE

- SMOTE algorithm (Synthetic Minority Over-Sampling Technique)
- Generates synthetic examples in the less frequent class
- For a randomly selected point $\vec{u}$, find a random nearest neighbor $\vec{w}$ within the *same* class
- Generate a new point $\vec{v}$ as

$$\vec{v} = \vec{u} + p \cdot (\vec{w} - \vec{u})$$

where $p \in (0, 1)$ is a uniform random number



Multi-Label Classification

- Objects may have multiple classes
- A news item may be both of business and politics
- Multi-label classification

Multi-Label Classification

- Objects may have multiple classes
- A news item may be both of business and politics
- Multi-label classification
- One-versus-rest classifiers
- For *m* original classes, *m* classifiers
- Need a special strategy if none of the classes is assigned

Multi-Label Classification

- Objects may have multiple classes
- A news item may be both of business and politics
- Multi-label classification
- One-versus-rest classifiers
- For m original classes, m classifiers
- Need a special strategy if none of the classes is assigned
- For m original classes, $2^m - 1$ combinations of classes
- Requires large training set