Data Mining & Machine Learning

Yong Zheng

Illinois Institute of Technology
Chicago, IL, 60616, USA

ILLINOIS TECH
College of Computing

Schedule

- Python Coding for Text Mining/Processing
- Semi-Supervised Learning

Semi-Supervised Learning

- Semi-supervised learning is an approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning and supervised learning
  - It is used to utilize unlabeled data to improve supervised learning
  - Or, it is used to utilize labeled data to improve unsupervised learning

Semi-Supervised Learning

- Categories by Applications
  - Semi-Supervised Classification/Regression
  - Semi-Supervised Clustering

Semi-Supervised Learning

- Categories by Applications
  - Semi-Supervised Classification/Regression
  - Semi-Supervised Clustering

Semi-Supervised Classification

Goal:

Using both labeled and unlabeled data to build better classifiers (than using labeled data alone).

Notation:

- input x, label y
- classifier $f : \mathcal{X} \mapsto \mathcal{Y}$
- labeled data $(X_l, Y_l) = \{(x_1, y_1), \ldots, (x_l, y_l)\}$
- unlabeled data $X_u = \{x_{l+1}, \ldots, x_n\}$
- usually $n \gg l$

Semi-Supervised Classification

- Solutions
  - Self-training
  - Co-training
  - Other methods….
- Note: these methods can also be applied to regressions

Self-training

Algorithm: Self-training

1. Pick your favorite classification method. Train a classifier f from (X_l, Y_l).
2. Use f to classify all unlabeled items $x \in X_u$.
3. Pick x^* with the highest confidence, add $(x^*, f(x^*))$ to labeled data.
4. Repeat.

The simplest semi-supervised learning method.

Self-training

Pros

- Simple
- Applies to almost all existing classifiers

Cons

- Mistakes reinforce themselves. Heuristics against pitfalls
  - 'Un-label' a training point if its classification confidence drops below a threshold
  - Randomly perturb learning parameters

Co-training

- Your data can be split into different views
- The view can be defined by different set of the features

Co-training

Each item is represented by two kinds of features
$x = [x^{(1)}; x^{(2)}]$

- $x^{(1)}$ = image features
- $x^{(2)}$ = web page text
- This is a natural feature split (or multiple views)

Co-training idea:

- Train an image classifier and a text classifier
- The two classifiers teach each other

Co-training

Algorithm: Co-training

1. Train two classifiers: $f^{(1)}$ from $(X_l^{(1)}, Y_l)$, $f^{(2)}$ from $(X_l^{(2)}, Y_l)$.
2. Classify X_u with $f^{(1)}$ and $f^{(2)}$ separately.
3. Add $f^{(1)}$'s k-most-confident $(x, f^{(1)}(x))$ to $f^{(2)}$'s labeled data.
4. Add $f^{(2)}$'s k-most-confident $(x, f^{(2)}(x))$ to $f^{(1)}$'s labeled data.
5. Repeat.

Co-training

Pros

- Simple. Applies to almost all existing classifiers
- Less sensitive to mistakes

Cons

- Feature split may not exist
- Models using BOTH features should do better

Semi-Supervised Learning

- Categories by Applications
  - Semi-Supervised Classification/Regression
  - Semi-Supervised Clustering

Semi-Supervised Clustering

- Clustering is an unsupervised learning process
- We can utilize labeled data to improve clustering
  - Amount of labeled data is limited

Semi-Supervised Clustering

- Input:
  - A set of unlabeled objects, each described by a set of attributes
  - A small amount of domain knowledge or labels
- Output:
  - A partitioning of the objects into k clusters
- Objective:
  - Maximum intra-cluster similarity
  - Minimum inter-cluster similarity
  - High consistency between the partitioning and the domain knowledge/labels
    - These knowledge/labels can be used as constraints
      - Must-Link = must be in a same cluster
      - Cannot-Link = must be in different clusters

Example: Semi-Supervised K-Means

- **Seeded K-Means:**
  - Labeled data provided by user are used for initialization: initial center for cluster *i* is the mean of the labeled data having label *i.*
  - Labeled data or Seed data are only used for initialization, and not in subsequent steps.
- **Constrained K-Means:**
  - Labeled data provided by user are used to initialize K-Means algorithm.
  - Cluster labels of seed data are kept unchanged in the cluster assignment steps, and only the labels of the non-seed data are re-estimated.

Schedule

- Python Coding for Text Mining/Processing
- Semi-Supervised Learning