Machine Learning HW4

Unsupervised Clustering & Dimensionality Reduction

TAs: ml2016ta@gmail.com

National Taiwan University

November 17, 2016

Dataset

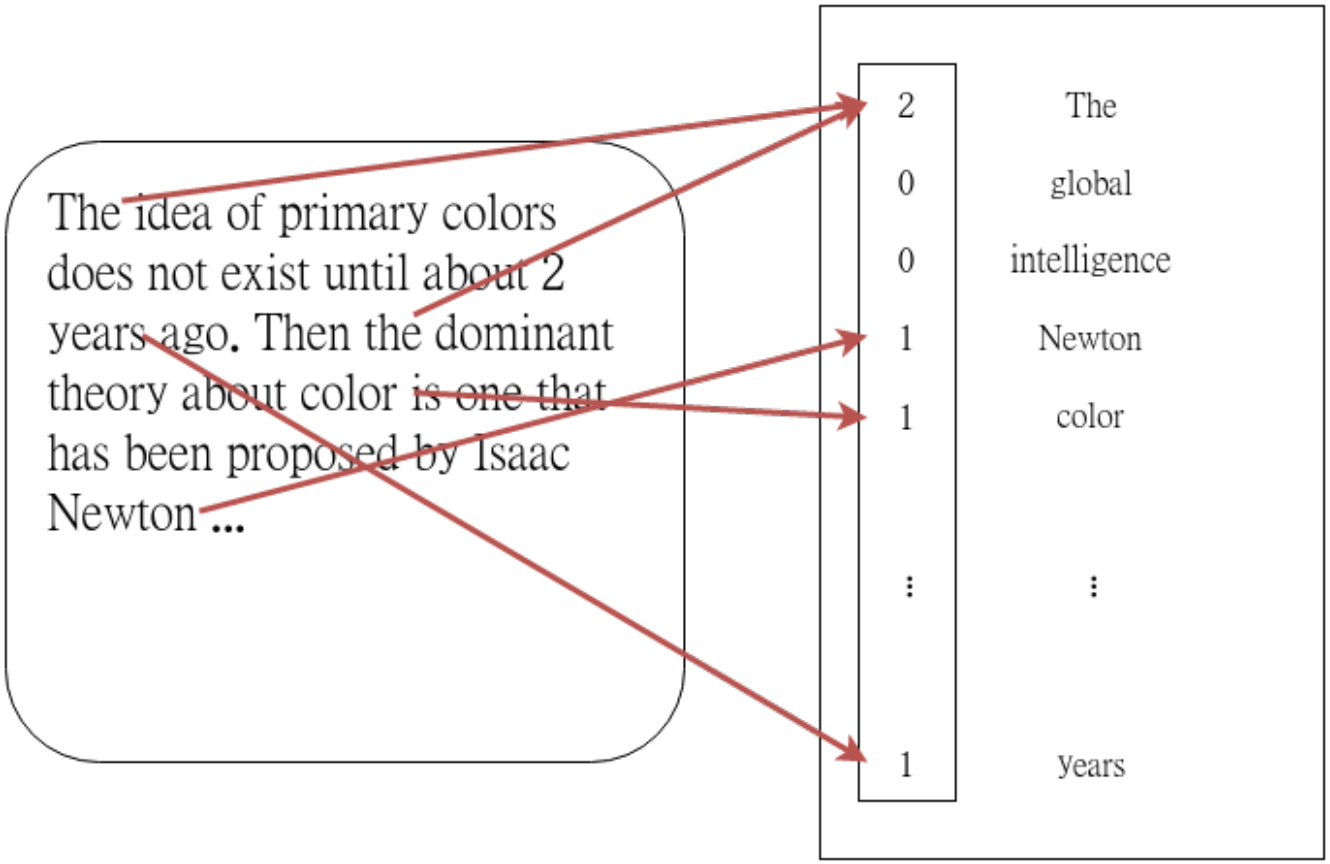
- Dataset: StackOverflow documents crawled from the internet.
- Original data (3M): each document has a title, document body, a few attributes, and multiple key term tags.
- $20,000$ documents were chosen from 20 different tags.
- Tags: wordpress, oracle, svn, apache, excel, matlab, visual-studio, cocoa, osx, bash, spring, hibernate, scala, sharepoint, ajax, qt, drupal, linq, haskell, magento.

Task Definition

- Objective: to group the documents into 20 clusters that correspond to the 20 given tags.
- External Evaluation: based on true labels
- F-Measure
  - A series of decisions ($\frac{n(n-1)}{2}$ pairs).
  - $P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}$.
  - $F_\beta = \frac{(1+\beta)^2 \cdot P \cdot R}{(\beta^2 \cdot P)+R}$.

Feature Extraction

- Bag-of-Words (BoW) features



- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

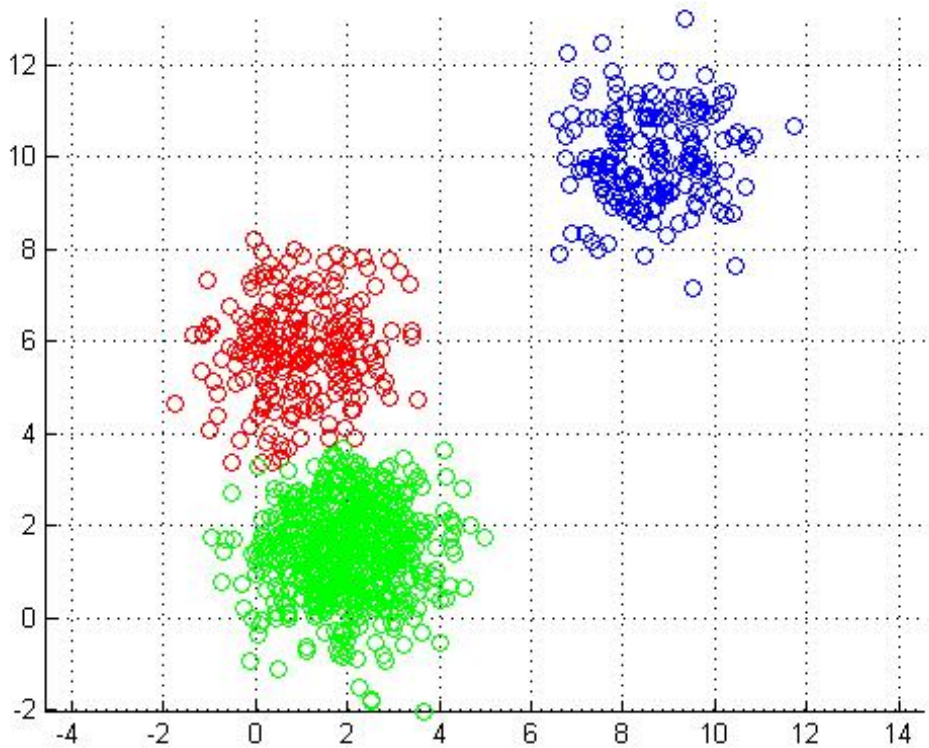
- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Feature Extraction

- Bag-of-Words (BoW) features
- Term frequency-inverse document frequency (TF-IDF)
  - TF: count the number of times each term occurs in each document (same as BoW).
  - IDF: diminishes the weight of terms that occur very frequently in the document set (e.g. "the", "and") and increases the weight of terms that occur rarely.
  - TF-IDF: TF $\times$ IDF ($w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$).
- Latent Semantic Analysis (LSA)
- Auto-Encoders
- Word Vectors/Document Vectors
- Note: try removing stopwords.

Clustering

- K-Means clustering



Data Visualization

- Principle Component Analysis (PCA)
- Truncated Singular Value Decomposition (truncated SVD)/LSA
- t-distributed Stochastic Neighbor Embedding (t-SNE)
  - Don't use on large d: SLOW; use PCA first if d is too large.
- We will provide labels after submission deadline for you to visualize the data.

Files

- Two files for training:
  - **title_StackOverflow.txt**: 20000 StackOverflow titles. Each line contains a title. $\mathrm{TitleID} = \mathrm{line\ number} - 1$.
  - **check_index.csv**: 5000000 test data pairs.
    ID: upload index
    x_ID: TitleID in title_StackOverflow.txt (start from 0)
    y_ID: TitleID in title_StackOverflow.txt (start from 0)
  - **docs.txt**: For training your paragraph/word vectors.
- One file for data visualization:
  - **label_StackOverflow.txt**: 20000 StackOverflow tags. Each line contains a tag id.
  - Will be released after Kaggle deadline.

Submission

- Format: .csv
- First row: ID, Ans
- Other rows: UploadID, Prediction
- UploadID: follow check_index.csv
- Prediction: 1 if in same cluster, 0 otherwise.
- Evaluation: $F_{0.25}$ ($\beta = 0.25$)

Kaggle

- Link: click here
- Login: NTU mail account
- This is an "individual" assignment.
- Team name: studentid_teamname. Do not use studentid if you are an auditor.
- Public/private sets: 50%/50%
- Kaggle deadline: 2016/12/02 09:00 (GMT+8).
- Release true labels: 2016/12/02 21:00 (GMT+8).
- Github deadline: 2016/12/02 21:00 (GMT+8).
- Report deadline: 2016/12/09 21:00 (GMT+8).

Scoring

- F-score on private set beats: (4%)
  - Weak baseline (2%)
  - Strong baseline (4%)
  - Top 3: bonus 1%
  - Note: if your code produces results that has too much discrepancy with your Kaggle results you will get 0%.
- Report: (5%)
  - Filename: report.pdf
  - 4 questions
  - 3 pages MAX, including figures.
  - Fontsize: **12pt**.
- Format: (1%)
  - Wrong script: 0%
  - Any other format error: 0.5% if you come to TAs and fix it.

Report Questions

1. Analyze the most common words in the clusters. Use TF-IDF to remove irrelevant words such as "the". (1%)
2. Visualize the data by projecting onto 2-D space. Plot the results and color the data points using your cluster predictions. Comment on your plot. Now plot the results and color the data points using the true labels. Comment on this plot. (1%)
3. Compare different feature extraction methods. (2%)
4. Try different cluster numbers and compare them. You can compare the scores and also visualize the data. (1%)

Rules

- No external data: You cannot use the true labels in any case. Do not pre-train word vectors/paragraph vectors with other data.
- Use any Python/C++ based packages: Keras, Tensorflow... (Torch included).
- No plagiarism.
- Late submission: 30% penalty per day (2 days max) **for each part**.

GitHub Repository

- Should include:
  - cluster.sh
  - report.pdf
- "cluster.sh" usage: ./cluster.sh $1 $2
  - $1: directory that contains "title_StackOverflow.txt" and "check_index.csv"
  - $2: output.csv
- Time limit: 5 minutes. (Upload your model if necessary).

Questions?