Machine Learning Homework 4

Unsupervised Learning & Dimensionality Reduction

ML TAs ntu.mlta@gmail.com

April 27, 2017

NTUEE

Outline

Eigenfaces with PCA

Visualization of Word Vectors

Estimation of Intrinsic Dimension

Summary and Policy

Eigenfaces with PCA

Introduction

Principle Component Analysis

In this exercise you will implement principle component analysis (PCA) to perform dimensionality reduction.

Dataset

You will be experimenting with the CMU AMP Lab facial expression database (link). The dataset contains 75 images for each of the 13 subjects.

Implementation

Implementing PCA

- First, you should center the data by subtracting the mean value of each feature from the dataset.
- Run SVD (preferred) on the centered data or eigen-decomposition on the covariance matrix to obtain the eigenvectors, which are also called eigenfaces.
- After computing the eigenfaces, you can use them to reduce the feature dimension.

Dimensionality Reduction with PCA

Projecting the data onto the principle components

- Given the dataset X, the principle components U, and the number of dimensions to reduce to k, you should project each face in X onto the top k components in U.
- Note that the top k components correspond to the k largest singular values.

Reconstructing the original faces with eigenfaces

After projecting the faces onto the lower dimensional space, you can approximately recover the original faces by projecting them back onto the original high-dimensional space.

Requirements

Follow the instructions below and answer the questions.

1. Perform PCA using the first 10 faces of the first 10 subjects to obtain the eigenfaces. Plot the average face. Also plot the top 9 eigenfaces in a figure (3-by-3, left to right & top to bottom). Show the figures in your report. (1%).
2. Project the 100 faces onto the top 5 eigenfaces, and then reconstruct the original images. Plot the 100 original faces (10-by-10) and the recovered faces (also 10-by-10). Show the two figures side-by-side in your report. (1%)
3. In problem 2, we can choose top k eigenfaces and check the reconstruction error (RMSE). Find the smallest k such that the error is less than 1%. (1%)

Restriction

You are only allowed to use `np.linalg.svd` or `np.linalg.eig` to calculate the eigenfaces. Libraries such as `scikit-learn` are prohibited.

Visualization of Word Vectors

Word2Vec

Introduction

In this exercise you will use Mikolov's renowned Word2Vec toolkit to train your own word vectors and visualize those vectors in 2-dimensional space.

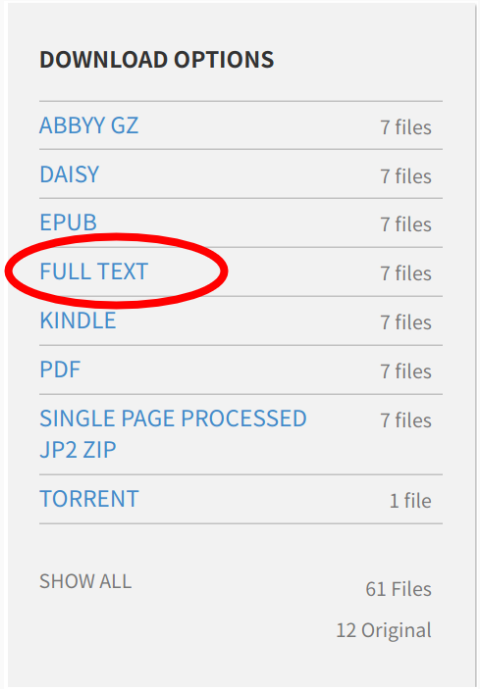
Toolkit

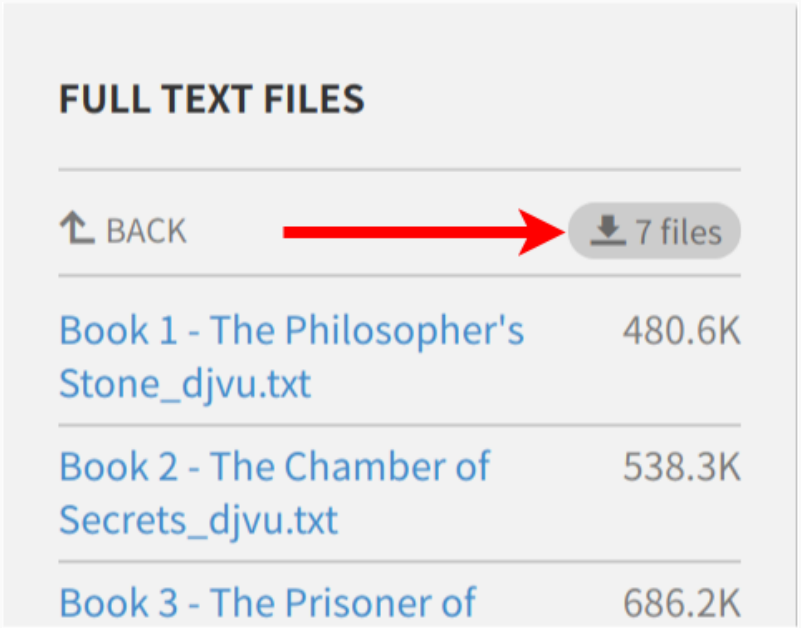
The original Word2Vec program was written in C, but a Python interface is available (link). A pre-compiled version is available on `pip`/`anaconda`.

Corpus: Harry Potter Series

Corpus

You will be using the Harry Potter series as your training corpus. Plaintext files are available on the internet (link). Please download the files by selecting “full text” and “7 files”.





After decompression, concatenate them into a single file (`cat * > all.txt`).

Word2Vec Training

Tutorial

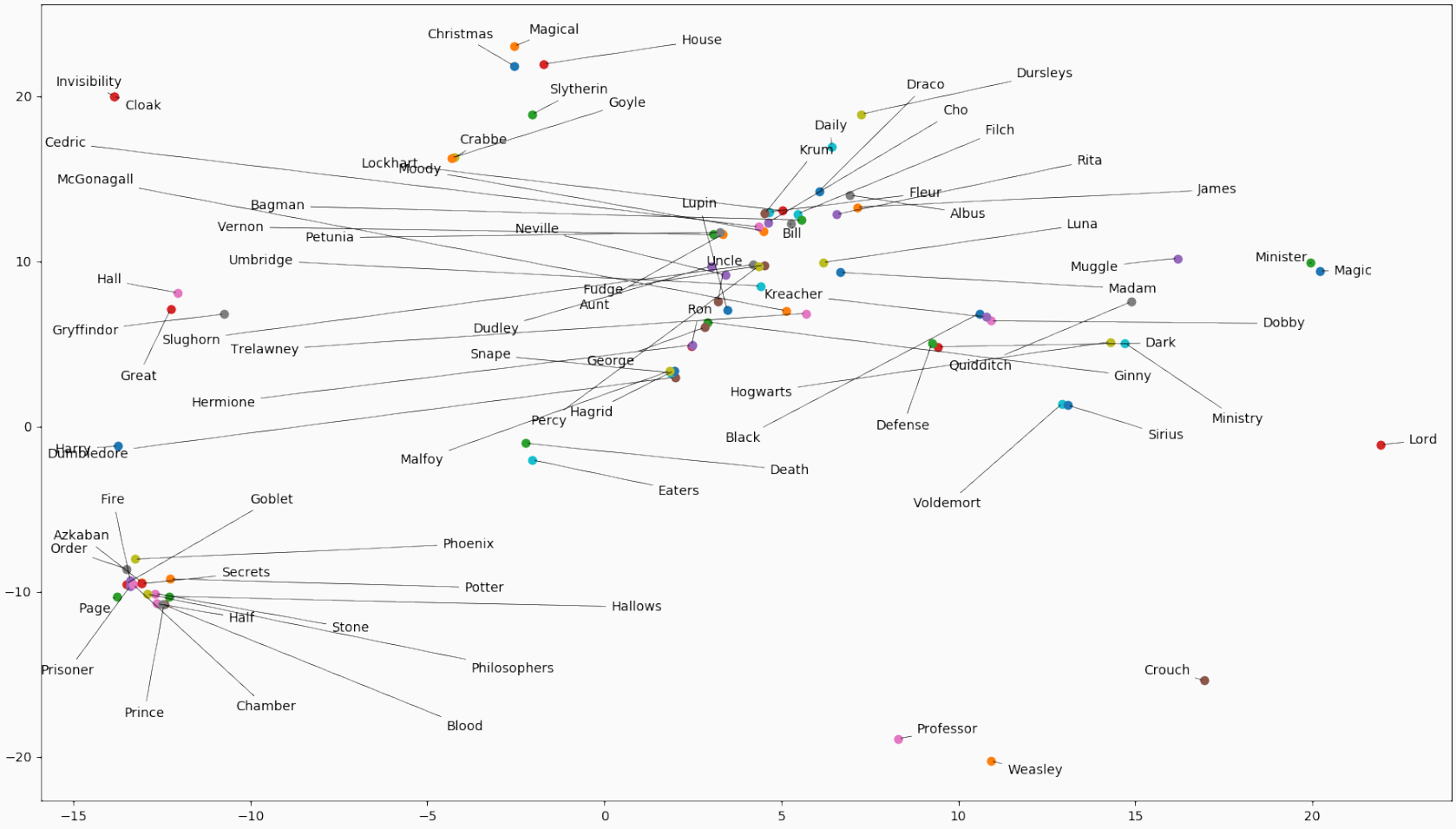
There is a iPython-notebook tutorial in the repository. You can try to tune the parameters to get better results in your experiments.

Tips: use `word2vec.word2vec` for training, `word2vec.load` to load your trained model, and `model.vocab` to obtain the vocabulary of the corpus in the order of word frequency.

Visualization Tips

1. Use `PCA` or `TSNE`(preferred) from the `scikit-learn` library to project the vectors of the k-most-frequent words (a value of 500$\sim$1000 is recommended).
2. Use the `NLTK` library to obtain the POS tags of the words. You should install `NLTK` and run `nltk.download()`. Choose "Models" and download:
   - `averaged_perceptron_tagger`,
   - `maxent_treebank_pos_tagger`,
   - `punkt`.
3. Plot the words that have tags `JJ`, `NNP`, `NN` and `NNS`, and ignore those that contain punctuations (" , . : ; ' ! ? ") or consist of only a single letter.
4. Use "adjust_text" (link) to prevent plot labels from overlapping.

Visualization Example



Requirements

Answer the following questions:

1. Train word vectors with the toolkit. Report the parameters you used and explain what the parameters mean. (1%)
2. Plot the visualization of word vectors on 2D space. Show the figure in your report. (1%)
3. Discuss your observations from the visualization. (1%)

Estimation of Intrinsic Dimension

Redundancy of Dimension

In the real world, the dimension of data we collected is often larger than the dimension of the underlying information. Redundant dimension has a bad influence on training models. So we need to find a way to represent our data in a compact way.

Difficulty

Representing data in a compact way without losing the information in data is difficult in general. But if we have some domain knowledge of our data, we may have some way to do it.

The Problem

There are 200 sets of data. Each set contains 10k-100k datapoints in $\mathbb{R}^{100}$. We know that the underlying dimension of these sets of data is much smaller than 100. In order to use these data effectively, we need to find the underlying dimension first...

Problem: Given the knowledge of these datasets in the next page, find the underlying dimension of each set.

Details

Let S_i denote the i-th set of data and d_i denote the underlying dimension of the data in S_i.

- $d_i \in [1, 60]$.
- The oracle network F_i of S_i is a neural network:

$$\mathbb{R}^{d_i} \xrightarrow{\text{ELU}} \mathbb{R}^{h_i} \xrightarrow{\text{ELU}} \mathbb{R}^{100} \xrightarrow{\text{Linear}} \mathbb{R}^{100}$$

  where h_i is a dimension sampled from $[60, 79]$ uniformly. Each layer performs a transformation $f(Wx + b)$, where W is a matrix, b is a vector, f is the activation function specified. All values of W, b are sampled from $\mathcal{N}(0, 0.5)$.
- The datapoints are sampled from $F_i(\mathcal{N}(0, I_{d_i}))$.
- Sample code to generate data $\rightarrow$ link.

Requirements

- You can use any toolkits.
- Upload your answer to Kaggle with the correct format. Note that if your answer for S_i is $\tilde{d}_i$, then you need to output $\ln \tilde{d}_i$. The evaluation method is Mean Absolute Error (MAE). (2%)
- In your report, please elaborate your method and why you used that method. Discuss the results in detail. (1%)
- Download the hand rotation sequence dataset (link), try to estimate the intrinsic dimension of this dataset and discuss your result in the report. (1%)

Kaggle

- Kaggle URL: link
- Please register your account with NTU mail.
- One student per team.
- Team name format: `(student_id)_.*`, do not prefix `student_id` in your team name if you are an auditor.
- Submission limit: 5 times per day.
- Test set is split into public set and private set, each contains 100 sets of data.

Format

Dataset

The dataset is a `.npz` file, it will be a `dict`-like object in Python after you load it with `np.load`.

Output

The output should be a `.csv` file containing two columns `SetId` and `LogDim`.

For more details, please refer to the documentation on Kaggle.

Appendix

ELU Activation

The ELU activation function is defined as:

$$\text{ELU}(x) = \begin{cases} x, & \text{if } x \geq 0, \\ e^x - 1, & \text{otherwise.} \end{cases}$$

Note that this function is continuous and 1-1.

Manifold Learning

In manifold learning, we must specify the dimension first. In this problem, we don't know the actual dimension, but the problem is also easier than finding the actual mapping from a high-dimensional space to a low-dimensional space.

Summary and Policy

Scoring & Report Problems

- Eigenfaces with PCA (3%)
  - Please refer to page 6.
- Visualization of Word Vectors (3%)
  - Please refer to page 13.
- Estimation of Intrinsic Dimension (4%)
  - Please refer to page 17 for the report problems. (2%)
  - Public/private score $\geq$ baseline before T_0 (0.5% each)
  - Public/private score $\geq$ baseline after T_0 [1] (0.5% each)
  - Kaggle rank top 5 (BONUS)

[1] $T_0 =$ 5/10 11:59:59 PM (GMT+8)

Policy I

Deadline

- Kaggle: 5/13 Sat. 11:59:00 PM (GMT+8)
- Report and source code: 5/14 Sun. 8:59:59 PM (GMT+8)

Policy II

Repository

- In your repository, `ML2017/hw4` should contain at least following:
  - `Report.pdf`
  - `pca.py`
  - `wordvec.py`
  - `dim.sh`
  - `requirements.txt`

  Also, include all code you use (except the Python packages).
- Don't upload extremely large files into the repository.

Policy III

Source Code

- Python Only. Use Python 3 if you can (3.6 is better). You can still use Python 2.7+ if you want.
- List packages you use in the `requirements.txt` (example). We will not handle any `ImportError` if that package is not listed.
- `dim.sh` should accept two arguments `$1`, `$2`.
  - `$1` is the path to a `.npz` file like the dataset on Kaggle.
  - `$2` is the path to the output `.csv` file.
  - Do NOT hard-code the file paths in your Python script.
  - Runtime limit: 10 minutes.
- !!! Please write your code in a readable way !!!

Policy IV

Report

- Please write the report in Chinese if you can (Although the slides are in English... :P). You can still write in English if you are more comfortable with it.
- Please convert your report into PDF format. (`.pdf`)
- Don't exceed 3 pages. (We would not grade the pages that exceed the limit)
- Template: link

Alert

Any form of plagiarism is prohibited. Cheating will result in zero points for this assignment. In addition, you may fail this course and may be reported to the school.

Questions?