Asmt 7: Dimensionality Reduction

Turn in through Gradescope by 11:59pm on Wednesday, April 5:
100 points

Overview

In this assignment you will explore regression techniques on high-dimensional data. **I am intentionally not providing you a link to the documentation to make you familiarize yourself with navigating the `sickit-learn` documentation.**

Report implementation only where asked.

Note: Homework assignments are intended to help you learn the course material, and successfully solve mid-term and final exams that will be done on paper in person.

As usual, it is recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. The latex source for this homework is available on Canvas. I recommend that you drop the two latex files (.tex and .sty) in a new Overleaf project, and compile/edit the .tex file there.

Submissions that are not uploaded on Gradescope will get 10% penalty.

1 Singular Value Decomposition (50 points)

You are going to use the following dataset of country-level child mortality and fertility in the year 2017. This data was downloaded from `https://www.gapminder.org/data/`. The `.csv` files are available on Canvas.

```
import pandas as pd
def prepare_data_problem_1():
    # Downloads from https://www.gapminder.org/data/
    cm_path = 'child_mortality_0_5_year_olds_dying_per_1000_born.csv'
    fe_path = 'children_per_woman_total_fertility.csv'
    cm = pd.read_csv(cm_path).set_index('country')['2017'].to_frame()/10
    fe = pd.read_csv(fe_path).set_index('country')['2017'].to_frame()
    child_data = cm.merge(fe, left_index=True, right_index=True).dropna()
    child_data.columns = ['mortality', 'fertility']
    child_data.head()
    print (child_data)

    return child_data
```

`child_data` consists of 186 examples (for 186 countries) and each example is represented with two features: mortality and fertility. Center the data and calculate the SVD of the centered data using `np.linalg.svd`.

(i) Report a scatter plot of 186 points and plot the two principal directions (rows of the matrix V^T). Consult this page for how to draw vectors: `https://stackoverflow.com/questions/42281966/how-to-plot-vectors-in-python-using-matplotlib`. Based on this plot, answer which direction seems more important and why?

(ii) Approximate the mortality/fertility dataset with the best rank-1 approximation (truncated SVD). Show the approximated data on the same scatter plot with the original data. You can use the following code for plotting:

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax1 = fig.add_subplot(111)
ax1.scatter(data['mortality'],data['fertility'], color='b')
ax1.scatter(approx_data['mortality'],approx_data['fertility'], color='r')
plt.show()
```

Answer (i):
Answer (ii):

2 Random Projection (50 points)

You are going to use the following code snippet to load the RCV1 dataset and downsample it to 500 examples. The full RCV1 contains over 800,000 newswire stories categorized in 103 classes. Each instance is represented with 47,236 features (cosine-normalized, log `TF-IDF` scores for each word in the vocabulary). More information is available here: `https://scikit-learn.org/stable/datasets/real_world.html#rcv1-dataset`.

```
import sklearn.datasets as dt
import random
random.seed(75)

def prepare_data_problem_2():
    '''
        Fetch and downsample RCV1 dataset to only 500 points.
        https://scikit-learn.org/stable/datasets/real_world.html#rcv1-dataset
    '''
    rcv1 = dt.fetch_rcv1()

    # Choose 500 samples randomly
    sample_size = 500
    row_indices = random.sample(list(range(rcv1.data.shape[0])),sample_size)
    data_sample = rcv1.data[row_indices,:]

    print(f'Shape of the input data: {data_sample.shape}') # Should be (500, 47236)
    return data_sample
```

Using `scikit-learn` apply:

- **A (25 points):** Gaussian random projection
- **B (25 points):** Sparse random projection

to the **downsampled** Reuters dataset for ε in the range from 0.1 to 0.99 with the step 0.2. Check a given value ε is acceptable with `johnson_lindenstrauss_min_dim` `sklearn.random_projection`, and if it is not, skip it. For each ε, use `sklearn.metrics.pairwise.euclidean_distances` to record a vector of the pairwise Euclidean distances of the actual data points, as well as a vector of the pairwise Euclidean distances of the transformed data points. Also calculate the mean of the vector of the absolute differences between the two vectors of pairwise distances.

(i) Report a plot with ε on the x-axis and the mean absolute differences on the y-axis.

(ii) Report a histogram of absolute differences.

(iii) Do your results in (i) and (ii) match the Johnson-Lindenstrauss lemma?

(iv) Report your implementation.

A:

Answer (i):
Answer (ii):
Answer (iii):
Answer (iv):

B:

Answer (i):
Answer (ii):
Answer (iii):
Answer (iv):

3 Bonus: Latent Semantic Analysis (LSA; 5 points)

LSA is basically the truncated SVD (with k equal to a desired number of topics) applied to a document-term matrix with TF-IDF scores. The matrix U_k obtained with SVD then represents a document-topic matrix and the matrix V_k is a term-topic matrix.

Implement LSA for the 20 newsgroups text dataset (`https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html`) and report your code. Set the number of topics to 10. From each right singular vector, identify and report the top-5 most important terms. Each one of these lists of 5 terms should hypothetically represent a topic. Do you see anything that resembles a topic? For example, this list `['film', 'films', 'movie', 'women', 'director']` indicates a topic of women in film.