COMS4060A/7056A: Assignment #3

Tim Bristow
tim@bristow.za.net

University of the Witwatersrand — November 9, 2021

Introduction

This assignment is based on content covering non-linear dimensionality reduction. You will be required to perform some basic data cleaning and exploration techniques on prescribed datasets. The aim is to explore the dataset and make observations. There is no strict requirement on the format of your submission, but any answers should be reasoned, discussed, and relevant data or results should be provided to substantiate this. You might like to submit either a PDF or a Jupyter notebook, for example. You can use any programming language or tool you would like, however.

Python Packages

You might find the following packages necessary/useful for this assignment if you are using Python (you will find many useful examples here too):

- Minisom: `https://github.com/JustGlowing/minisom`
- umap-learn: `https://umap-learn.readthedocs.io/en/latest/`
- astroML: `https://www.astroml.org/`

You can install them using:

```
$ pip install umap-learn minisom astroML
```

1 Astronomy Data

You will need to implement several dimensionality reduction techniques to view low-dimensional projections of galaxy & quasar spectra from the Sloan Digital Sky Survey `https://www.sdss.org/`. Sample notebooks are provided, with *some* of the code provided (many **IMPLEMENT ME**s for you to complete).

Your goal is to find projections that provide a separation of classes of spectra, such that the visualisations might allow intuitive evaluation of the relationships between points.

Some background on the dataset and a discussion of PCA on the spectra is provided as an attached PDF: *2.3.6. Dimensionality Reduction of Astronomical Spectra.pdf*. The notebook you will be working off performs PCA on the spectra, and shows that at least 20 PCs are needed to retain sufficient variance.

Perform the following manifold methods and compare the differences:

- UMAP
- Modified LLE
- Spectral Embedding (sklearn's implementation of Laplacian Eigenmaps)
- ISOMAP

In particular, you should investigate the effect of:

- dataset size
- number of clusters
- number of neighbours
- the impact of the above on the 2 or 3D embedding (ie visualisation) - does it show a visual difference between classes?

You should address the following questions for each of the methods. You will probably find that you need to take an iterative approach and move back and forth between these questions. All of the parameters will affect the others in some way, so choose a suitable baseline for each, and then iterate. For all sections show 2/3d plots of the embedding to motivate your answer.

Remember to normalise your data. **[1 mark]**

1.1 Dataset size [6 marks]

Change the number of data points used to find the projection. Choose a *random* sample of the data of sizes, say, 50%, 75% and 100% of the total dataset.

1. How stable is the projection between different subsets of the data? **[4 marks]**
2. Which of these manifold methods appears to give the most stable results? **[2 marks]**

1.2 Number of neighbours [10 marks]

All of these methods consider a neighbourhood of points when building up their adjacency matrices. The neighbourhood size is set with the n_neighbours parameter and can have a large impact on the output of the algorithm.

1. How does the number of neighbors change the projection? **[1 mark * 4]**
2. Which of the manifold methods appears to have the most stable results as the number of neighbors is changed? **[2 marks]**
3. Plots to support your conclusions. **[1 mark * 4]**

1.3 Number of components [12 marks]

For all of these methods, you can adjust the number of components, which is the output dimension of the algorithm.

1. How does the number of components change the projection? **[1 mark * 4]**
2. Which of the manifold methods appears to have the most stable results as the number of components change? **[2 marks]**
3. Plots to support your conclusions. **[1 mark * 4]**
4. There is code in the notebook which performs part of the Laplacian Eigenmapping solution by hand (note, it uses a normalised Laplacian with sparse matrix methods, similar to what is performed by SpectralEmbedding in sklearn). Plot the leading number eigenvalues (from which you might find a suitable number of clusters). How does this compare with your findings when looking for the most stable result? **[2 marks]**

This should give you a projection of the data that gives a good visualisation of the relationship between points. An astronomer may go further and try to develop rough cut-offs that would give a broad classification to an unlabeled test point. This sort of procedure could be used as the first step of a physically-motivated classification pipeline, or to flag potentially interesting objects for quick followup.

Total for Question 1: 29 marks

2 Gene Expression Data

You are given gene expression data for 14 types of cancer. Each gene expression signature has 16063 genes. There are 144 training and 54 test samples from patients xwith one of these types of cancers.

The datasets are attached for your convenience, but you can also download them here `https://web.stanford.edu/~hastie/ElemStatLearn/data.html`.

- Dataset description: 14cancer.info
- Dataset: 14cancer.xtrain, 14cancer.xtest
- Dataset Labels: 14cancer.ytrain. 14cancer.ytest

You need to investigate this data using a variety of techniques: self-organising map, UMAP, and LLE. Before starting, remember to normalise your data **[1 mark]**.

We would also like to go further than just plotting, and see if these low-dimensional representations are retaining useful information from the original data (could we use the embedding as input into an additional model?). Since we have labelled data, one way we can do that is to perform clustering and see which labels are being assigned correctly. We will use the adjusted mutual information score `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html` to quantify this.

2.1 SOM [10 marks]

Start with the SOM. There are a number of parameters you need to set for the SOM:

- the map size
- the neighbourhood size (set through the sigma parameter in minisom)
- the learning rate
- number of iterations used in training

There is a classify() method in the attached notebook which will take some test data and classify each datapoint by mapping it to a node in the embedding. You can compare this with the actual label of the data and see the adjusted mutual information score. Try a few iterations with training parameters and look at how it affects the performance. **[5 marks]** When you have set up your parameters and performed the training, produce the following plots:

1. Plot the weights of the map (som.get_weights()) **[1 mark]**
2. Plot the U-Matrix (som.distance_map()) **[1 mark]**
3. Plot the clusters (som.winner()) **[1 mark]**
4. From the above results, can you identify any clusters? Which cancer types cluster together? **[2 marks]**

(There is an example of a U-matrix/cluster plot at the end of this assignment sheet.)

2.2 UMAP and LLE [10 marks]

Now, compare this with UMAP and LLE:

1. Perform UMAP and LLE on the same data and plot the leading two components of their embeddings. Comment on the outputs. **[2 * 2 marks]**
2. Perform k-means (with 14 clusters) on the output embeddings from UMAP and LLE. Plot the clusters (with their predicted labels), and calculate the adjusted mutual information score. **[2 * 2 marks]**

3. From these results, which dimensionality reduction method would you advise? LLE, UMAP, or SOM? Motivate your answer. **[2 marks]**

For Reference and to learn more about the data and previous results using SOMs on gene data:

- S. Ramaswamy et al. (2001), Multiclass Cancer Diagnosis Using Tumor Gene Expression Signatures, *Proc. Natl. Acad. Sci.*, 98, p15149-15154, 2001. URL: `https://pubmed.ncbi.nlm.nih.gov/11742071/`.
- McGarry K., Sarfraz M., MacIntyre J. (2007) Integrating Gene Expression Data from Microarrays Using the Self-Organising Map and the Gene Ontology. In: Rajapakse J.C., Schmidt B., Volkert G. (eds) Pattern Recognition in Bioinformatics. PRIB 2007. Lecture Notes in Computer Science, vol 4774. Springer, Berlin, Heidelberg. URL: `https://link.springer.com/content/pdf/10.1007/978-3-540-75286-8_21.pdf`.

Total for Question 2: 21 marks

Submission

Work by yourself or in groups of up to four people. Submit your work to Moodle as a PDF or Jupyter notebook.

After discussions with the class, I will only be including the marks for the top 2 assignments of the 3.

Deadline: end of November 2021 (sooner is better, as it gives more time for marking before the exam))

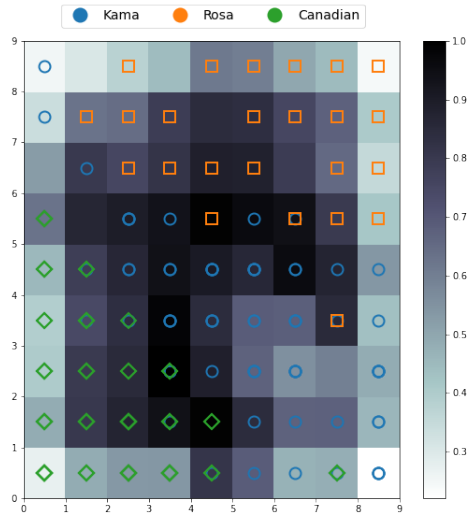

Figure 1: Example of a U-Matrix plot with winning classes highlighted for each node in the SOM map.