Randomized Optimization

Assignment 3

CS 7641

N. Ryan Karel (nkarel3)

# Introduction

We will be examining the business classification (ADD CITATION FOR BUSCLASS) and wine quality datasets previously seen in my first assignment for this class (Newman, Hettich, Blake, & Merz, 1998). As a quick reminder, these datasets principally vary in terms of their shapes. The business classification data has many more records and many more features (1,000 to be precise). Each feature here pertains to the presence of a single word in a natural language description of the business. In contrast, the wine dataset is shorter in length and only has twelve features, including values such as sulfate content and pH. The labels here are wine quality. As a reminder, there are 13 distinct classes captured in the business classification data, and only 3 for the wine dataset. Given the drastically differing sizes of the datasets, we should expect substantially different results when performing dimensionality reduction and clustering between them.

# Step 1

We begin with two clustering algorithms: K-Means and the Expectation-Maximization. We will first seek an appropriate number of clusters for each algorithm on each of our datasets in a completely unsupervised way: we do not want to consider the corresponding labels of records or clusters in this first step. After we have committed to a particular number of clusters, we will consider labels to help evaluate the utility of our clusters.

## Determining the number of clusters

## Cluster evaluation

### K-Means

When reviewing the resulting clusters’ general labelling, we see some encouraging trends on both of these datasets. When tuning the K-Means algorithm for our business classification task, we might have found the optimal number of clusters unsatisfying. Given that there are thirteen possible labels but only two clusters, is some signal slipping through the cracks? When we review which labels are in the two clusters, we find that the K-Means algorithm was finding meaningful separation in the data, just not the one that our labels would dictate. Take a look at the following table.

(INSERT KM BUSCLASS TBL HERE)

We see that Cluster 0 seems to include a greater proportion of white collar industries (e.g. Financials, IT) while Cluster 1 seems to hold a greater proportion of blue collar industries (e.g. Industrials, Materials). This was exciting to see, as it provided some proof that despite the usage of labels in the fitting process, the K-Means algorithm was able to split the data up in a way that was understandable to humans.

We see similar results from the wine classification K-Means clusters by reviewing which qualities are present in which clusters. Four clusters may have seemed overkill for a dataset with only three possible labels, but there is still a trend in the more prevalent labels of each cluster.

(INSERT KM WINE TBL HERE)

Although it struggled with the medium quality wine, the clusters are able to help separate the low and high quality items fairly well.

### Expectation-Maximization

Not unlike K-Means, The E-M algorithm did suggest that 2 clusters were optimal for the business classification data, though it did not produce as interpretable trends in the corresponding labels. The only rationale I can think of to justify this result is the fact that the algorithm is more equipped to deal with ambiguity (REVIEW THE RESULTING PROBABILITIES TO VALIDATE) through its emphasis on probability of belonging to a particular Gaussian process, and hence there are more data points that the algorithm estimates could truly go either way.

(INSERT ARTIFACTS TO BACK THIS UP)

In contrast, the wine quality clusters show a more pronounced trend than the corresponding results of the K-Means algorithm.

(INSERT WINE E-M TBL HERE)

This could be due simply to the fact that we are using three clusters instead of four, and hence better matching the distribution of the various wine qualities.

# Step 2

We now shift focus to our dimensionality reduction techniques, including PCA, ICA, Random Projections, and Linear Discriminant Analysis.

## PCA

Given the width of the business classification dataset and the expected correlations between much of its features, I fully expected the PCA algorithm to dramatically reduce the number of columns and still explain a large portion of the variance of the columns. Strangely, this did not happen. Instead, the algorithm still needed to use just over 850 features to capture 95% of the variance within this dataset. A mere 15% reduction was certainly unexpected, which led me to believe that the columns were not nearly as correlated as I had suspected.

Chart

Description automatically generated

This stands in contrast to the PCA algorithms performance on the wine dataset, on which it was able to capture a substantial amount of variance (defined as being over 95%) with only 3 components.

Chart

Description automatically generated

The fact that the algorithm could explain effectively 100% of the variance with only five components caused me to wonder if we could perfectly reconstruct the dataset using just these six components. This did appear to be the case. Why were we able to do this? When we review the individual features for the wine dataset, we do see values that should be heavily correlated, for example acidity measures and pH, residual sugar and alcohol, et cetera.

Chart, line chart

Description automatically generated

In summary, to achieve our requisite explained variance ratio, we could only reduce the business classification dataset by 15% of its columns but an entire 75% for the wine dataset.

## ICA

# References

Hayes, G. (2019). *mlrose: Machine Learning, Randomized Optimization and SEarch package for Python.* Retrieved from https://github.com/gkhayes/mlrose

Isbell, C., Viola, P., & De Bonet, J. (1997). *MIMIC: Finding Optima by Estimating Probability Densities.* MIT, Cambridge. Retrieved from https://www.cc.gatech.edu/~isbell/papers/isbell-mimic-nips-1997.pdf

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). *UCI Repository of machine learning databases*. Retrieved from http://www.ics.uci.edu/~mlearn/MLRepository.html

scikit-learn. (2022). *sklearn.neural\_network.MLPClassifier*. Retrieved from scikit-learn.org: https://scikit-learn.org/stable/modules/generated/sklearn.neural\_network.MLPClassifier.html