Classification of Weather Data

Kieran B. A. Molloy
Faculty of Science and Technology
Lancaster University
UK
k.molloy@lancaster.ac.uk

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

II. Pre-Processing

The initial step of a classification problem is to investigate the provided dataset, ensuring it is clean. Additionally some of the methods applied in later sections require data to be in a specific format, such as within an interval of $\mathbf{x} \in [0, 1]$. For the dataset at hand, the first step is to check for missing values of which there are none. Additionally there are no categorical values, hence no one-hot encoding or alterior methods are required. The second step is transforming the data to a known plane using normalisation, standardisation or equivalent techniques. Standardisation over the interval $[-1, 1]$ was attempted, but caused problems with other algorithms, as did normalisation, and so a simple standardisation over $[0, 1]$ resulting in column-wise, $\overline{x} = 0$ and $\sigma = 1$. Additionally, feature binarisation was considered for the snow, and rain indicators however this was later replaced by data subsetting (see Section III) as this was found to give better clustering performance. Furthermore, K-Bins discretisation for the sunshine variable was able to reduce variability and increase clustering performance too.

III. Feature Selection

Data is often represented in high dimensional matrix form, but many features are often redundant, noisy or correlated, [1]. Which can result in poor performance or over fitting. This problem can be resolved by reducing the number of features being considered, there are various strategies, with varying results and speed [2]. As described in [3], feature selection, from a label availability perspective, can be categorised into Supervised Feature Selection and Unsupervised Feature Selection. The dataset considered requires unsupervised feature selection as there are no ground truth labels. Initially the dataset is split on multiple conditions, the first is snowfall. Splitting on the condition of $x > 0$ to create two datasets, df_standard and df_snowing. The standard subset is then split again for rainfall, on the condition $x > 0$ creating df_raining and df_standard.

A. Supervised Feature Selection (SFS)

Supervised Feature Selection methods, such as, [2] [4] , are generally able to efficiently output good features due to having access to training labels, which contain all information required to classify.

B. Un-supervised Feature Selection (UFS)

However in un-supervised feature selection, training labels are not available, which makes feature selection far more difficult. The most prominent methods are Unsupervised Discriminative Feature Selection (UDFS) [5], which aims to select the most discriminative features for data representation. And Non-negative Discriminate Feature Selection (NFDS) [6], which performs non-negative spectral analysis and feature selection simultaneously. Both of these methods ignore data cleanliness, outliers and noise. Which leads to the Robust Unsupervised Feature Selection (RUFS) [3], which performs robust clustering and robust feature selection simultaneously to determine which features hold the most information. In this problem Principal Feature Analysis (PFA) will be used [7]. It is chosen to be used due to its simple implementation, similar criteria to Principal Component Analysis (PCA) (described in further detail in Section IV), which maximises the variability of the features in the lower dimensional space and minimising the reconstruction error. The PFA method is briefly described below. Let X be a zero mean n-dimensional random feature vector. Let Σ be the covariance matrix of X. Let A be a matrix whose columns are the orthogonal eigenvectors of the matrix Σ

$$\Sigma = A \Lambda A^T \tag{1}$$

$$\text{where} \quad \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \ddots & 0 & \\ & 0 & \ddots & \\ & & & \lambda_n \end{bmatrix} \tag{2}$$

$\lambda_1, \ldots, \lambda_n$ are the eigenvalues of Σ and $A^T A = I_n$. Let A_q be the first q columns of A and let $V_1, \ldots, V_n$ be the first rows of the matrix A_q. Each vector V_i represents the projection of the i'th feature of the vector X to the lower dimensional space. Creating a PFA algorithm from scratch, also using the standard PCA from sklearn (as the L1-Norm algorithm described above caused compatibility issues) it can be seen that when reducing the dataset to 6 dimensions, the highest influence dimensions are 0, 7, 2, 3, 10, 14 respectively.

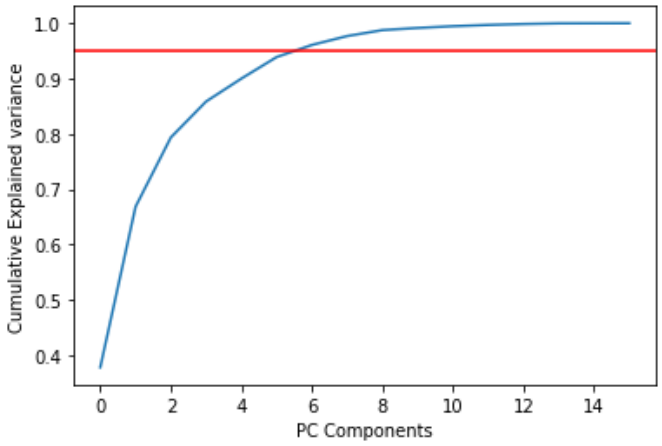
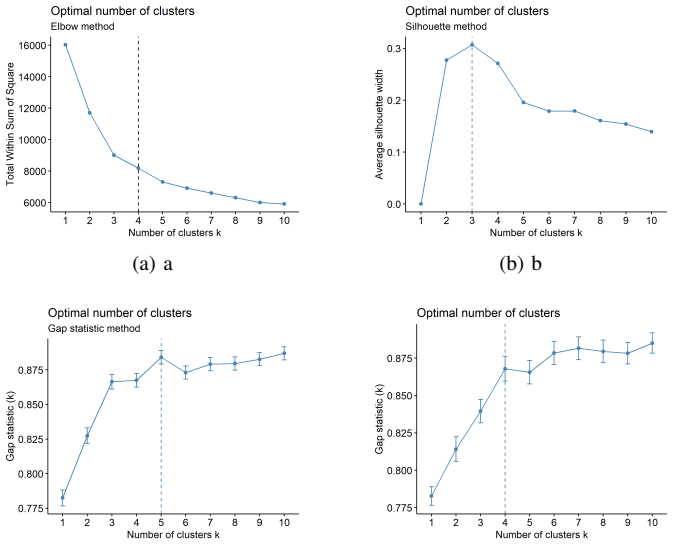


Fig. 1. Explained Variance for increasing pc components, with 95% marked in red

IV. Principal Component Analysis

Principal Component Analysis (PCA) is often used during exploratory data analysis and for creating predictive models by leveraging dimensionality reduction by projecting data points into lower principal components. The principal components of a set of points in real space are a sequence of p direction vectors, where the i-th vector is the direction of a line that best fits the data whilst being orthogonal to the first $i-1$ vectors. It is notable that the principal components are eigenvectors of the covariance matrix, and as such can be computed using eigendecomposition. An additional benefit of using PCA is the low computing cost due to using singular-value decomposition (SVD). However in modern big data sets, faulty or corrupted data, often referred to as 'outliers', can cause sensitivity problems. This is due to the L2-PCA places squared emphasis on the magnitude of each data point coordinate, which over-emphasises peripheral points (the outliers that are easier to detect and remove). For this reason, instead of using a standard PCA or L2-PCA, the L1-Norm formulation places linear emphasis on the data point coordinates which is more robust, and invariant to rotations [8]. The L1-Norm is more computationally expensive, with $\mathcal{O}(2^{NK})$ where K is the principal component rank, and $X \in \mathcal{R}$ for a DxN matrix. This can be minimised to $\mathcal{O}(N^{DK-K+1})$ when $d = \text{rank}(X)$ and $K < D$. However, using a novel algorithm utilising bit flipping the L1-Norm calculation can be reduced to something closer to standard L2-Norm PCA calculations $\mathcal{O}\big(NDmin(N,D) + N^2K^2(K^2+d)\big)$ for all $K < d$. This algorithm is demonstrated to be superior to standard L2-Norm PCA for characterising potentially faulty data [9]. The algorithm described in this paper is implemented from scratch using just SVD function from scipy package as this benefits from the optimisation in C.

Figure 1 shows the explained variance curve as a cumulative sum of explained variance ratios for increasing i principal components, for a 90% level 5 components are sufficient and for 95% 6 components are sufficient, 6 PC's will be used for later analysis.



(a) a (b) b (c) c (d) d

Fig. 2. df_standard cluster analysis; (a) demonstrates the elbow method (b) demonstrates the silhouette method (c) demonstrates the gap statistic method with $ns = 25$ and $nboot = 500$ (d) demonstrates the hierarchical clustering method with $kmax = 10$ and $nboot = 500$

V. Clustering

As the dataset contained no ground truth labels, unsupervised clustering must be performed and evaluated.

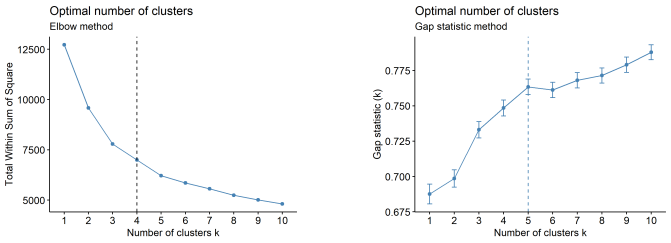
A. Cluster Analysis

One of the fundamental problems with clustering is determining the optimal, k, number of clusters. There are a number of methods for measuring similarities and estimating the optimal clusters. Generally these fall into two categories:

1) Statistical Methods : Comparing evidence vs null hypothesis
2) Direct Methods : Optimising a criterion

The first method described is the elbow method, which looks at the within-cluster sum of square (WSS), and attempts to select a k value whereby an additional cluster improves little. The second method is the silhouette approach, which computes the average silhouette for different k values and seeks to maximise [10]. The third method which is generally considered the most accurate, the gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be the value that maximises the gap statistic. [11]. As described in Section III the data is split into 3 datasets, labelled df_raining, df_snowing and df_standard.

Fig 2 demonstrates various algorithms for the standard dataset. The elbow method is suggesting 4 clusters is appropriate, the silhouette and hierarchical method is suggesting 3 where as the gap statistic is suggesting 5. All 3 of these values will be tested for accuracy. Fig 3 demonstrates the elbow and



(a) a (b) b

(c) c (d) d

Fig. 3. (a), (b) Consider df_raining for the elbow and gap statistic method (c),(d) Consider df_snowing for the elbow and gap statistic method. N.B the gap statistic method uses $ns = 25$ and $nboot = 500$

gap statistic methods for the raining and snowing subsets, the gap statistic for the snow subset suggests using 1 cluster is sufficient, and for the raining subset 5 is recommended. The elbow method suggests 4 for both, which is an expected result due to how its calculated. Additionally the silhouette method suggests 2 for both snowing and raining subsets. Hence 5 clusters will be used for the raining subset, and 1 cluster will be used for the snowing subset and will be given the label "snowing" and no longer considered for clustering.

B. Cluster Evaluation

Evaluating the performance of a clustering algorithm can be complicated, especially without ground truth labels, and as such the only comparisons that can be made are using the model itself. This paper considers 3 scoring methods; Silhoette Coefficient (CS) , Calinksi-Harabasz Index (CHI) and Davies Bouldin Index (DBI). The silhouette coefficient s for a single sample is given as

$$s = \frac{b - a}{\max(a, b)}$$

where a is the mean distance between a sample and all other points in the same class, and b is the mean distance between a sample and all other points in the nearest cluster [12]. The score is bounded between $[-1, 1]$ for incorrect clusters to high dense clusters, with higher scores when clusters are dense and well separated. However sometimes it can misperform when using density based clusters (such as DBSCAN). The Calinkski-Harabasz index (Variance Ratio Criterion), score s for dataset E of size n_E which has been clustered into k clusters.

$$s = \frac{\text{tr}(B_k)}{\text{tr}(W_k)} \cdot \frac{n_E - k}{k - 1}$$

where $\text{tr}(B_k)$ is thr trace between the group dispersion matrix and $\text{tr}(W_k)$ is the trace of the within-cluster dispersion matrix defined by

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

and

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

where C_q is the set of points in cluster q, c_q is the centre of cluster q, c_E is the center of E, and n_q is the number of points in q. A high Calinkski-Harabasz score generally relates to dense and well seperated clusters and is fast to compute, however it suffers the same fate as the silhouette score and can misperform with convex clusters such as density based clustering [13]. The final evaluation metric is the Davies-Bouldin Index s for

$$s = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

where

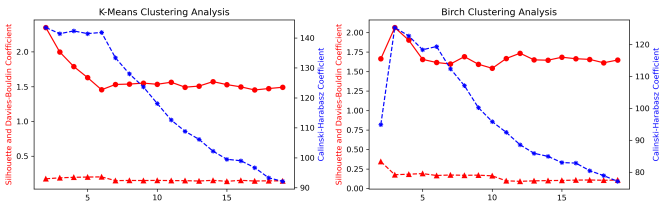
$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

. s_i is the average distance between each point of cluster i and the centroid as of that cluster, d_{ij} is the distance between cluster centroids i and j. A condition implied is that R_{ij} is non-negative and symmetric. The Davies-Bouldin index is the average similarity between clusters, and 0 is the lowest possible score which relates to well defined clusters and it is fast to compute but it suffers the same fate as the previous two with convex clusters and misperforms. Another large drawback for this method is the limitation to euclidean space due to the usage of centroid distances [14] [15].

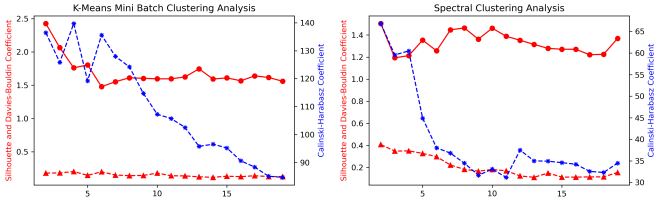
This paper will compare 4 different clustering methods:

1) K-Means
2) Birch
3) Mini Batch K-Means
4) Spectral Analysis

Figures 4 and 5 demonstrate the performance of the clustering algorithms using the evaluation methods defined above. Notably in Figure 4 K-Means, Birch and Mini-Batch KMeans seemingly follow a quadratic-like curve with Spectral Analysis being more 'random'. The clustering scores on the standard dataset peak at 4,5 using the CHI metric for all clustering methods, however a general downward trend can be seen for all metrics as the number of clusters increases. This can also be seen for the raining dataset, Figure 5, additionally the silhouette score does not see much change for varying clusters

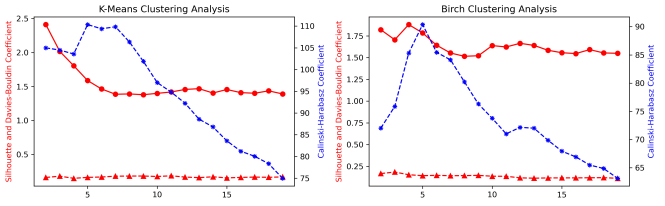


(a) a (b) b

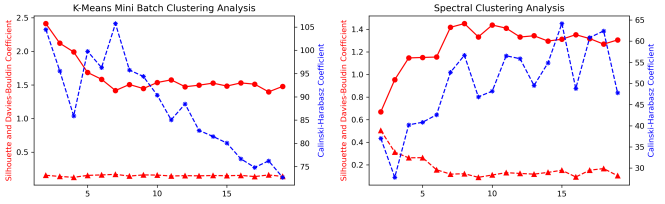


(c) c (d) d

Fig. 4. (a), (b) Consider df_raining for the elbow and gap statistic method (c),(d) Consider df_snowing for the elbow and gap statistic method. N.B the gap statistic method uses $ns = 25$ and $nboot = 500$



(a) a (b) b



(c) c (d) d

Fig. 5. (a), (b) Consider df_raining for the elbow and gap statistic method (c),(d) Consider df_snowing for the elbow and gap statistic method. N.B the gap statistic method uses $ns = 25$ and $nboot = 500$

VI. Classification

VII. Conclusion

VIII. References

References

[1] H. Liu, X. Wu, and S. Zhang, "Feature selection using hierarchical feature clustering," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 979–984. [Online]. Available: https://doi.org/10.1145/2063576.2063716

[2] T. Phyu and N. Oo, "Performance comparison of feature selection methods," *MATEC Web of Conferences*, vol. 42, p. 06002, 01 2016.

[3] M. Qian and C. Zhai, "Robust unsupervised feature selection," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, 2013, p. 1621–1627.

[4] S. Zheng and C. Ding, "A group lasso based sparse knn classifier," *Pattern Recognition Letters*, vol. 131, pp. 227 – 233, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865519303940

[5] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou, "l2, 1-norm regularized discriminative feature selection for unsupervised learning," in *IJCAI*, 2011.

[6] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, ser. AAAI'12. AAAI Press, 2012, p. 1026–1032.

[7] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," in *Proceedings of the 15th ACM International Conference on Multimedia*, ser. MM '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 301–304. [Online]. Available: https://doi.org/10.1145/1291233.1291297

[8] N. Kwak, "Principal component analysis based on l1-norm maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1672–1680, 2008.

[9] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient l1-norm principal-component analysis via bit flipping," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4252–4264, 2017.

[10] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, ser. Wiley Series in Probability and Statistics. Wiley, 2009. [Online]. Available: https://books.google.co.uk/books?id=W2zOYtmKovIC

[11] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00293

[12] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0377042787901257

[13] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[14] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[15] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2, pp. 107–145, Dec 2001. [Online]. Available: https://doi.org/10.1023/A:1012801612483