TECHNICAL REPORT

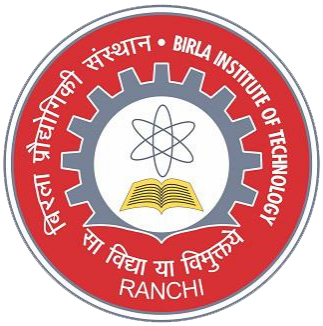
Grouping of Medicinal Drugs used for Similar Symptoms by Mining Clusters from Drug Benefits Reviews

by

Apurva Bhargava

(BE/25022/15)

BE-CSE-VII



Birla Institute of Technology, Mesra

Contents

Grouping of Medicinal Drugs used for Similar Symptoms by Mining Clusters from Drug Benefits Reviews

Apurva Bhargava (BE/25022/15), BE-CSE-VII, Birla Institute of Technology, Mesra, Jaipur

1. Introduction

Clustering is an unsupervised machine learning algorithm used to group multi-dimensional data-set into closely related groups or clusters, in such a way that objects in the same cluster are more similar to each other than to those in the other clusters. It is a main task of exploratory data mining, and a common technique for statistical data analysis, and has applications in computational biology, bioinformatics, medicine, business and marketing, World Wide Web, etc.

Problem Statement: The Drug Review Dataset [1] provides the benefit reviews of patients on specific medicinal drugs along with related conditions and names of drugs. The objective is to analyse the patients' reviews and group them into clusters, in order to identify which drugs were used for similar conditions and similar benefits. The information furnished through this cluster analysis can be used for understanding implicit relations in a group of drugs and benefits or for checking possibility of suggesting alternative medicine prescription for a given condition.

2. Data Description and Data Preprocessing

2.1 Dataset

The Drug Review Dataset (Druglib.com) provides patient reviews on specific drugs along with related conditions. Furthermore, reviews are grouped into reports on the three aspects benefits, side effects and overall comment. Additionally, ratings are available concerning overall satisfaction as well as a 5 step side effect rating and a 5 step effectiveness rating. The data was obtained by crawling online pharmaceutical review sites. [1] The tab-separated values (TSV) file (test data) contained the following attributes:

Attribute	Type	Details
urlDrugName	Categorical	Name of drug
condition	Categorical	Name of condition
benefitsReview	Text	Patient's review on benefits
sideEffectsReview	Text	Patient's review on side-effects
commentsReview	Text	Overall Patient comment
Rating	Numerical	Rating out of 10 by patient

sideEffects	Categorical	5 step side-effect rating
effectiveness	Categorical	5 step effectiveness rating

2.2 Preprocessing

- The only attributes relevant to the problem statement were benefitsReview (for generating features for cluster analysis), urlDrugName (for finding the drug name using the index of the benefitsReview in a given cluster), and condition (for finding symptoms using the index of benefitsReview and also for validating cluster analysis using visualization methods, as discussed later). The rest of the attributes were discarded.
- Inconsistent and incomplete examples were removed. The final working dataset contained 895 examples.

```
example_corpus = [
    'My health improved significantly and pain was reduced.',
    'I instantly felt a reduction in inflammation.’,
    'Significant improvement and instant relief.',
    'Did not work for me.'
]
```

- Each patient review was converted to lower case.
- Each patient review was cleaned by removing certain stopwords (a, an, the, and, but, etc.), since they have very high frequency, but are not good descriptors (noise).
- Words in patients’ reviews were reduced to their word stems. Even if the stems are no longer actual words, it can be reasonably expected that most of their different forms are reduced to the same stem. For example, tries, tried and try- all reduce to tri.

```
preprocessed_corpus = [
    'my health improv signific pain was reduced. ',
    'i instant felt reduc in inflammation. ',
    'signific improv instant relief. ',
    'did not work for me. '
]
```

- Other suggested text preprocessing steps are removal of punctuations, expansion of negatives, lemmatization, POS-tagged based cleaning, and removal of sparse (low-frequency) words. These were not used considering the small size of dataset and the short length of reviews.

3. Feature extraction

Clustering algorithms like K-means require numerical inputs for computation. In order to extract informative numerical features from the textual data (benefitsReview), a normalized TF-IDF (Term Frequency- Inverse Document Frequency) representation is created. TF-IDF is a numerical statistic, a sort of weighting factor that is intended to reflect how important a word is to a document in a collection or corpus.[2]

The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. The feature map thus generated is a good numeric representation of all the reviews in the dataset, and can be used to understand the similarity between any two reviews in terms of vectors containing feature value for every word in the corpus.

Firstly, the corpus of text reviews is converted to a matrix of token counts (frequency of each token in the given review, for all reviews), called the count matrix.

```
feature_names = ['did', 'felt', 'for', 'health', 'improv', 'in', 'infl
ammation', 'instant', 'me', 'my', 'not', 'pain', 'reduc', 'reduced', '
relief', 'signific', 'was', 'work']

count_matrix = [
    [0 0 0 1 1 0 0 0 0 1 0 1 0 1 0 1 1 0]
    [0 1 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0]
    [0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0]
    [1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1]
]
```

The count matrix is transformed to a normalised TF-IDF representation.

$$tfidf(d,t) = tf(t) \times idf(d,t)$$

$$tf(t) = count\_matrix[d][t]$$

$$idf(d,t) = \log\left(\frac{n+1}{df(d,t)+1}\right) + 1$$

where tfidf(d,t) = tfidf value for d^{th} review's term t,

tf(t) = term frequency of term t,

idf(d,t) = inverse document frequency of d^{th} review's term t (the constant "1" is added to the numerator and denominator of the idf as if an extra document was seen containing every term in the collection exactly once, which prevents zero divisions),

df(d,t) = number of reviews containing term t = sum of the column for term t in count_matrix,

n = total number of reviews [3], [4]

```
tfidf_features = [
    [0.         0.         0.         0.40801493 0.28949879 0.
     0.         0.         0.         0.40801493 0.         0.40801493
     0.         0.40801493 0.         0.28949879 0.40801493 0.       ]
    [0.         0.47122483 0.         0.         0.         0.47122483
     0.47122483 0.3343481  0.         0.         0.         0.
     0.47122483 0.         0.         0.         0.         0.       ]
    [0.         0.         0.         0.         0.44782471 0.
     0.         0.44782471 0.         0.         0.         0.
     0.         0.         0.63115694 0.44782471 0.         0.       ]
    [0.4472136  0.         0.4472136  0.         0.         0.
     0.         0.         0.4472136  0.         0.4472136  0.
     0.         0.         0.         0.         0.         0.4472136]
]
```

4. Clustering Model, Algorithm and Validation Measures

4.1 Model employed

k-means clustering is a method of vector quantization, popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum.

4.2 Algorithm

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the k-means algorithm; it is also referred to as Lloyd's algorithm.

Given an initial set of k means (marked as centroids), $m_1^{(1)}$, $m_2^{(1)}$, …, $m_k^{(1)}$, the algorithm proceeds by alternating between two steps:

I. *Assignment Step:* Assign each observation to the cluster whose mean has the least squared Euclidean distance, this is intuitively the "nearest" mean.

$$S_i^{(t)} = \{x_p : \left\|x_p - m_i^{(t)}\right\|^2 \leq \left\|x_p - m_j^{(t)}\right\|^2 \forall \, j, 1 \leq j \leq k\}$$

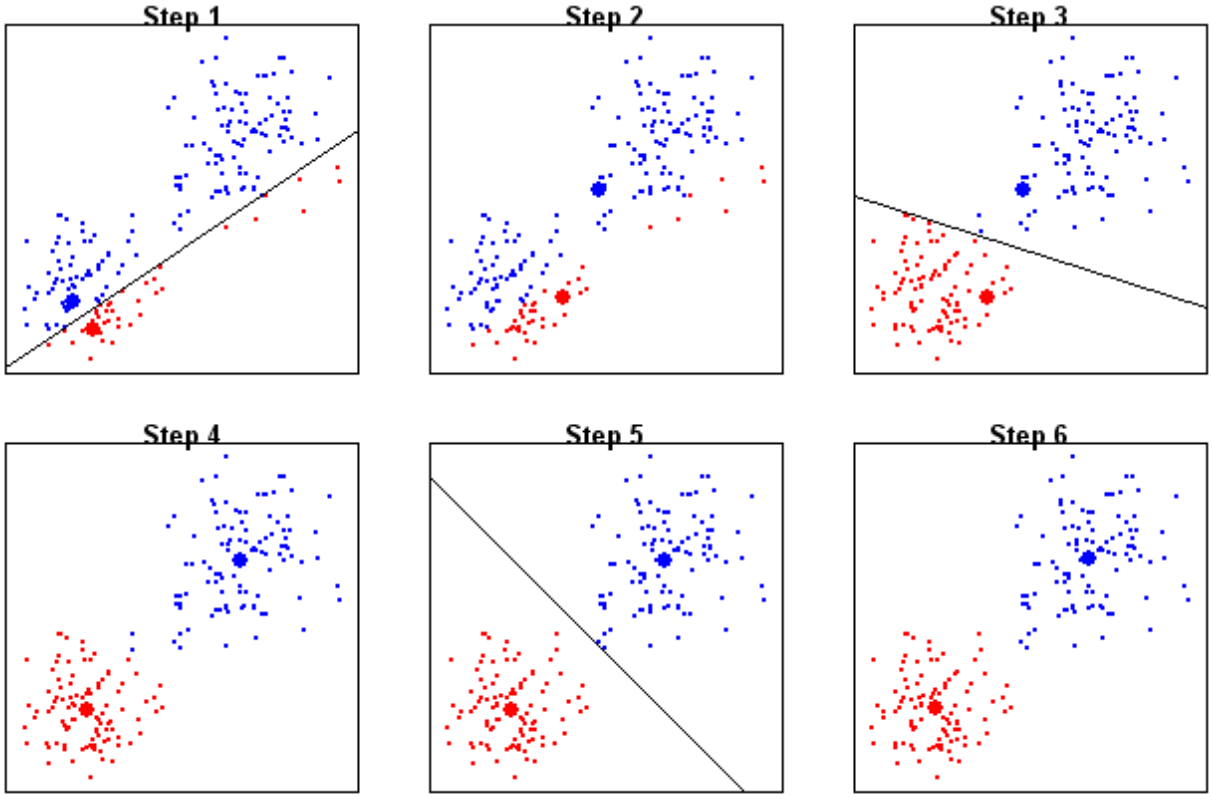
where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

II. *Update step:* Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm has converged when the assignments no longer change. There is no guarantee that the optimum is found using this algorithm. [5]

The following diagram illustrates the computation of new mean, and the observations assigned to the corresponding cluster at each step of the algorithm:



4.3 Measures for validation

Since clustering is an unsupervised learning method, some validation measure is required to check the effectiveness of the clustering model.

4.3.1 Silhouette Score

Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The silhouette coefficient is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance. The former has been used here.

$$SC^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max(a^{(i)}, b^{(i)})}$$

$$SS_{overall} = \frac{\sum_{i=0}^{n-1} SC^{(i)}}{n}$$

where $SC^{(i)}$ is the Silhouette Coefficient of observation i,
$SS_{overall}$ is the overall Silhouette Score (also lies between -1 and 1),
n = number of observations,
$a^{(i)}$ = mean intra-cluster distance of observation i, and
$b^{(i)}$ = smallest mean nearest-cluster distant (of which i is not a member). [6]

4.3.2 Calinski and Harabaz Score

Calinski and Harabaz score, also known as the Variance Ratio Criterion is defined as the ratio between the within-cluster (intracluster) dispersion and the between-cluster (intercluster) dispersion. Higher score indicates better clustering.

$$CH = \frac{\sum_i \frac{n_i d^2(c_i, c)}{(NC - 1)}}{\sum_i \sum_{x \in C_i} \frac{d^2(x, c_i)}{(n - NC)}}$$

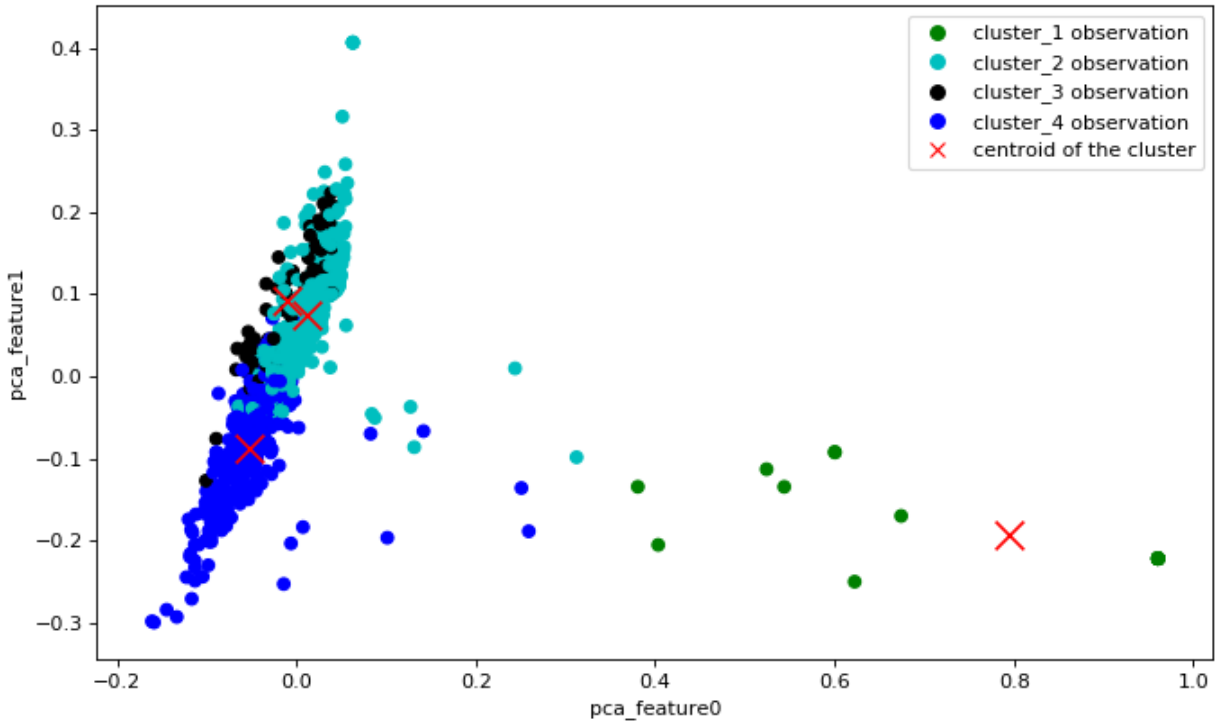
where CH is the overall Calinski and Harabaz score of dataset D,
n = number of observations in D,
c = center of D,
NC = number of clusters,
C_i = ith cluster,
n_i= number of observations in the cluster C_i,
c_i= center of cluster C_i, and
d(x, y) = distance between x and y. [7]

5. Results and Output

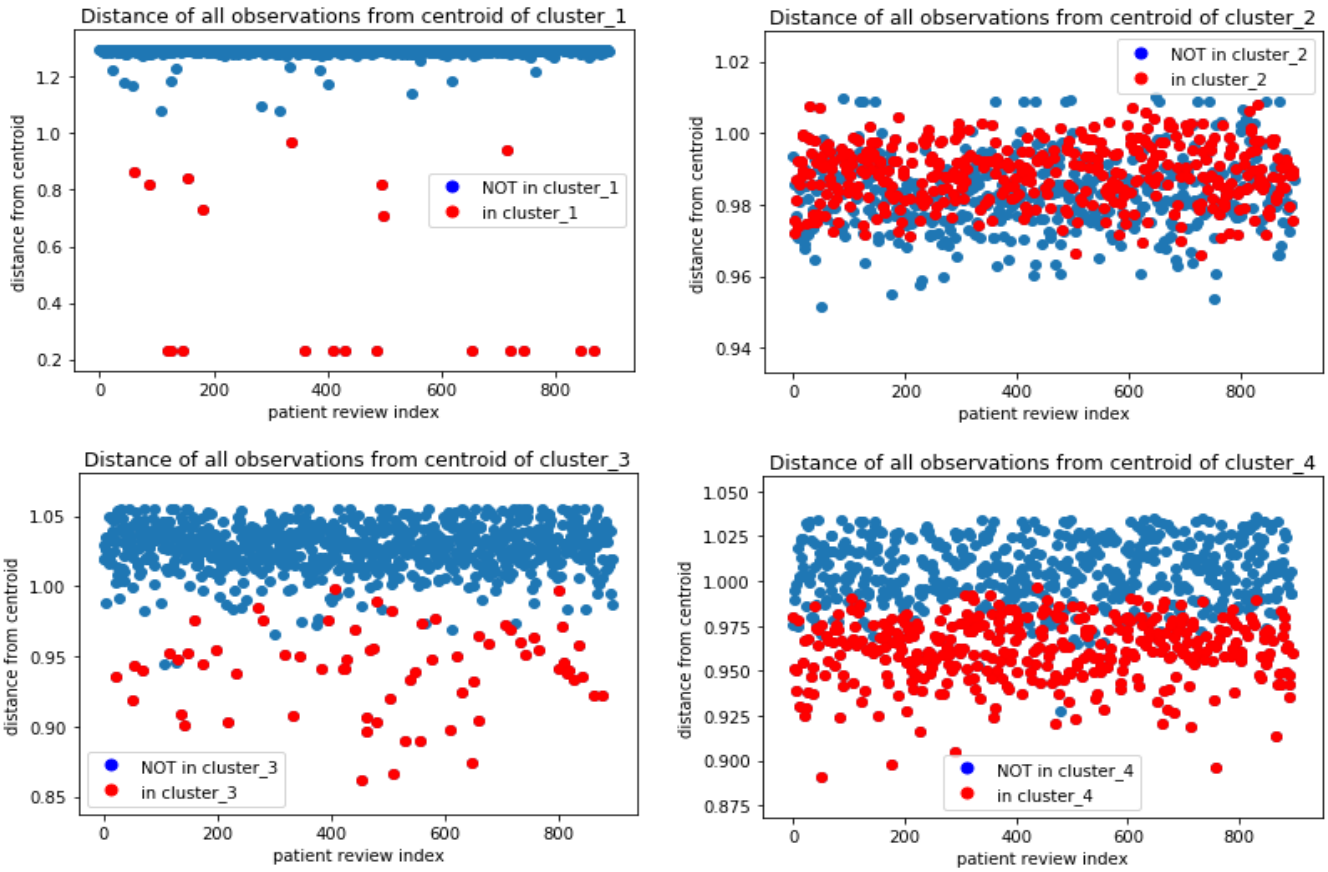
Lloyd's algorithm for k-means clustering was used on feature map of dimensions 895x3614 and the number of clusters was set to 4 for ease of validation and visualisation of results.

Output cluster	Number of Observations
cluster_1	20
cluster_2	426
cluster_3	68
cluster_4	381

A. Using Principal Component Analysis (PCA), two representative features were identified from the tfidf feature map and the observations and cluster centroids were plotted:



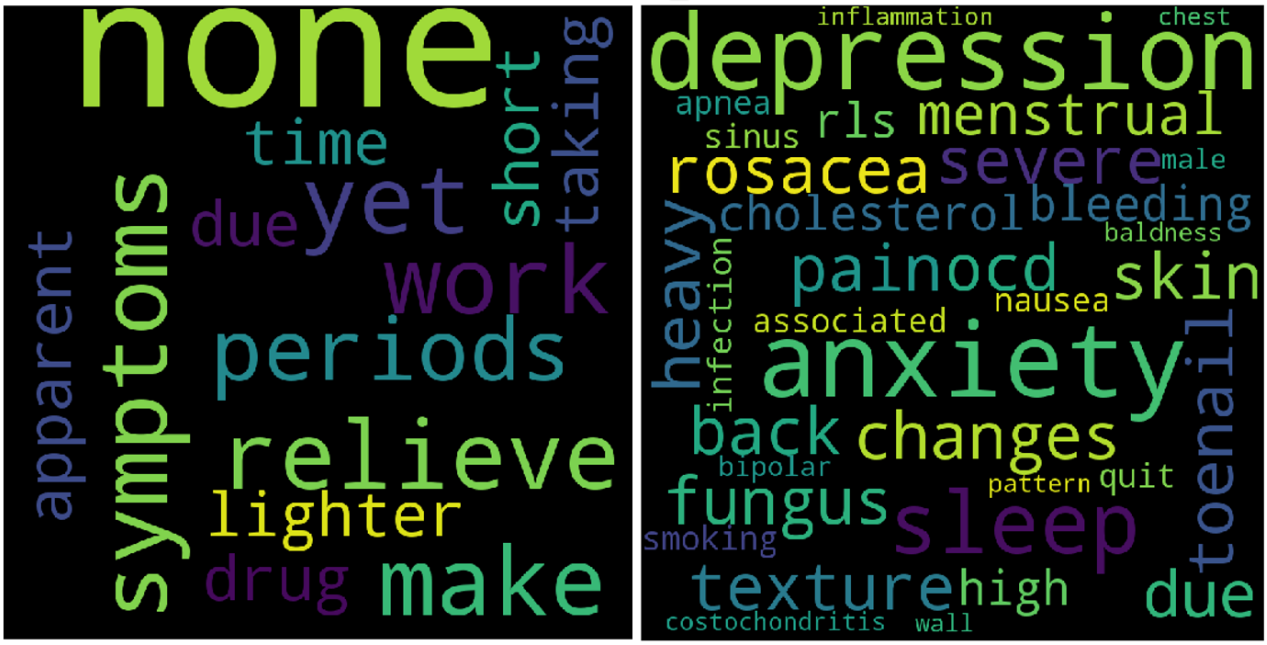
B. The following scatter plots clearly show that the cluster includes the points that are at minimum distance from the cluster's centroid:



C. Visualising context similarity in clusters using Word Cloud:
Word Clouds are a great tool for visual representation of the term frequencies in a corpus. The words of larger size occur more frequently than the words of smaller size. And the words that frequently occur together are closer. The word cloud on the left is that of patients' reviews (benefitsReview) corpus and that on the right is that of the corresponding conditions (condition) for which the drugs were used.

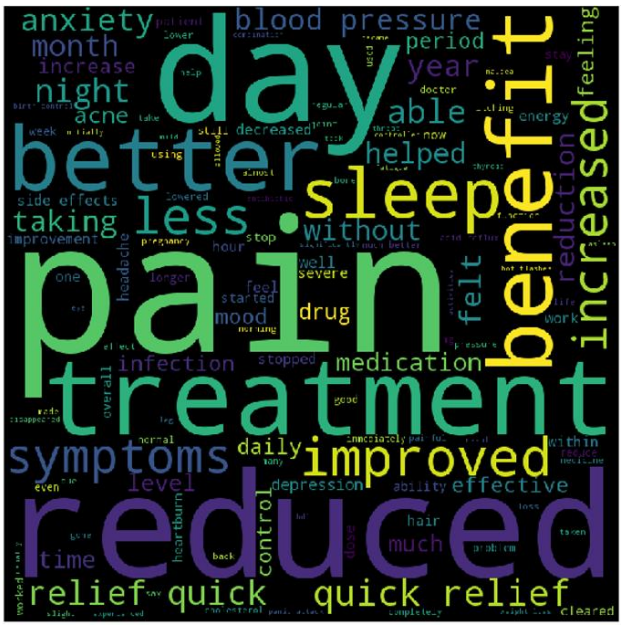
cluster_1 Analysis: There are only 20 reviews in cluster_1, including those which only say "None". There isn't much similarity to be discerned from such a small cluster. However certain words show time-related or temporal words, for example, 'due', 'yet', 'time', 'short'.

benefit review : cluster_1 : condition



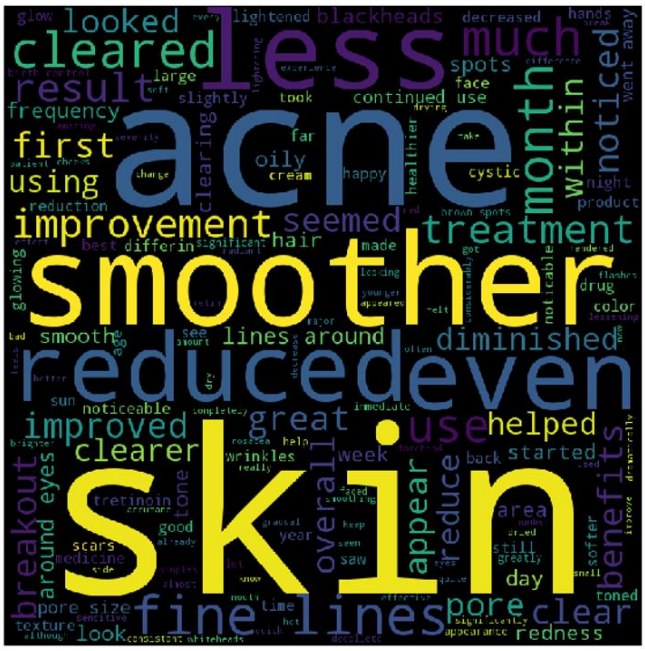
cluster_2 Analysis: There are 426 reviews in cluster_2. From the review word cloud, the most frequently used words are ‘reduced’, ‘treatment’, ‘better’, ‘pain’, ‘benefit’, ‘improved’, ‘quick relief’, etc. which have positive implications. Words like ‘day’, ‘night’, ‘month’, ‘year’, ‘daily’ show that the associated drugs had to be taken regularly. The word cloud of conditions highlights ‘depression’, ‘anxiety’, ‘bipolar’, ‘disorder’, ‘insomnia’, all of which are neurological or psychological issues, thus hinting to regular use of medicinal drugs. The other most commonly identified conditions that are correlated are ‘blood pressure’, ‘high blood’, ‘cholesterol’, ‘hypertension’, ‘heartburn’, ‘diabetes’, etc.

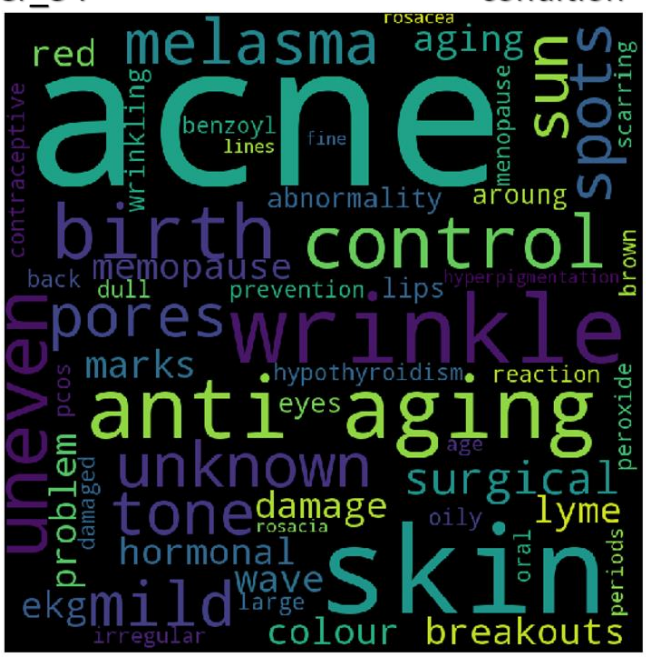
benefit review : cluster_2 : condition



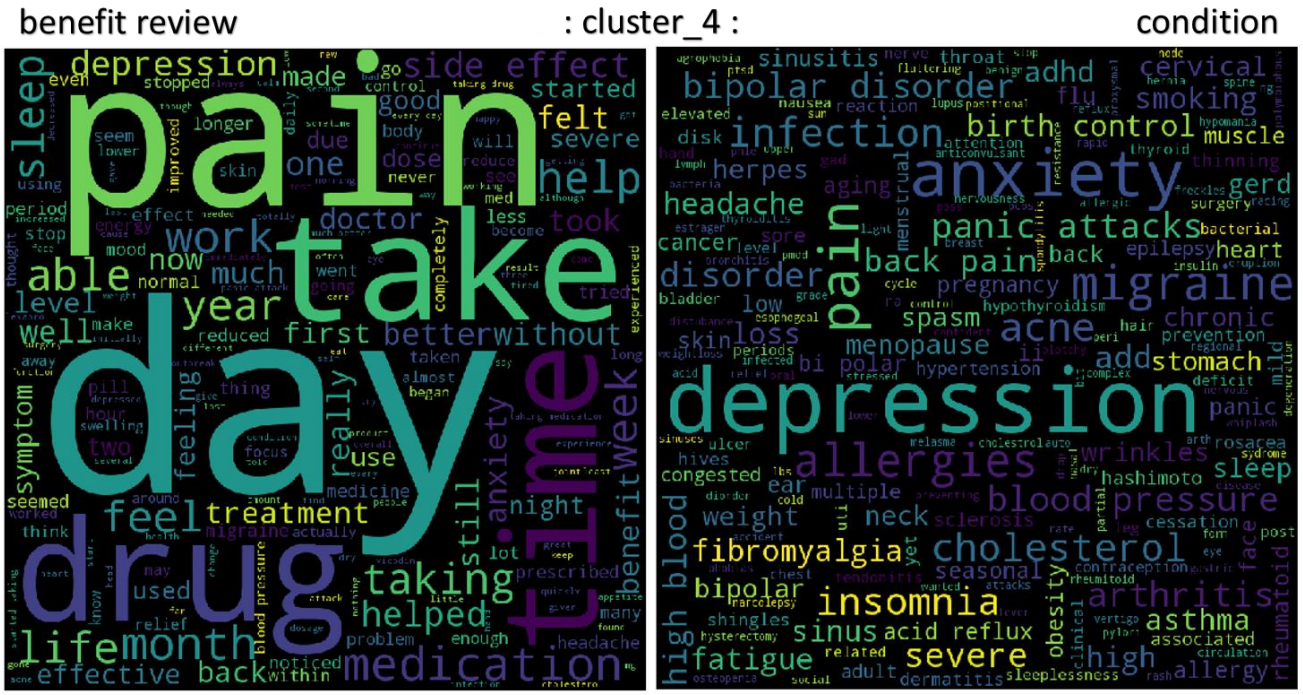
cluster_3 Analysis: cluster_3 contains 68 reviews. The most common words, according to review word cloud, are ‘skin’, ‘smoother’, ‘acne’, ‘less’, ‘reduced’, ‘even’, ‘cleared’, ‘fine’, ‘lines’, ‘pore’, ‘breakout’, ‘blackheads’, ‘appear’, etc. The most frequent words in condition are ‘acne’, ‘wrinkle’, ‘skin’, ‘anti’, ‘aging’, ‘birth’, ‘control’, ‘sun’, ‘spots’, ‘pores’, ‘uneven’, ‘hormonal’, etc. All of these point to face-care, skin-care or cosmetic drugs. This cluster is very specific, and indicates that 60-70 reviews is an ideal cluster size for this analysis.

benefit review : cluster_3 : condition





cluster_4 Analysis: 381 reviews are grouped together. Frequent words in reviews are ‘day’, ‘pain’, ‘take’, ‘drug’, ‘time’, ‘medication’, ‘month’, ‘year’. Frequent words in conditions are ‘depression’, ‘anxiety’, ‘bipolar’, ‘disorder’, ‘panic’, ‘attacks’, ‘migraine’, ‘insomnia’, ‘adhd’, etc. Owing to a large number of reviews, there is lack of specificity, however, there are a lot of reviews on neurological and psychological drugs.



D. Overall Silhouette Score = 0.01801918102027343
Since the score is greater than 0, the clustering algorithm worked well to define clusters. However, the score is not close to 1, which suggests overlapping among clusters.

E. Calinski and Harabaz Score = 9.803

6. Comparison with two other alternatives

6.1 ALTERNATIVE (1) Agglomerative (Hierarchical) Clustering

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

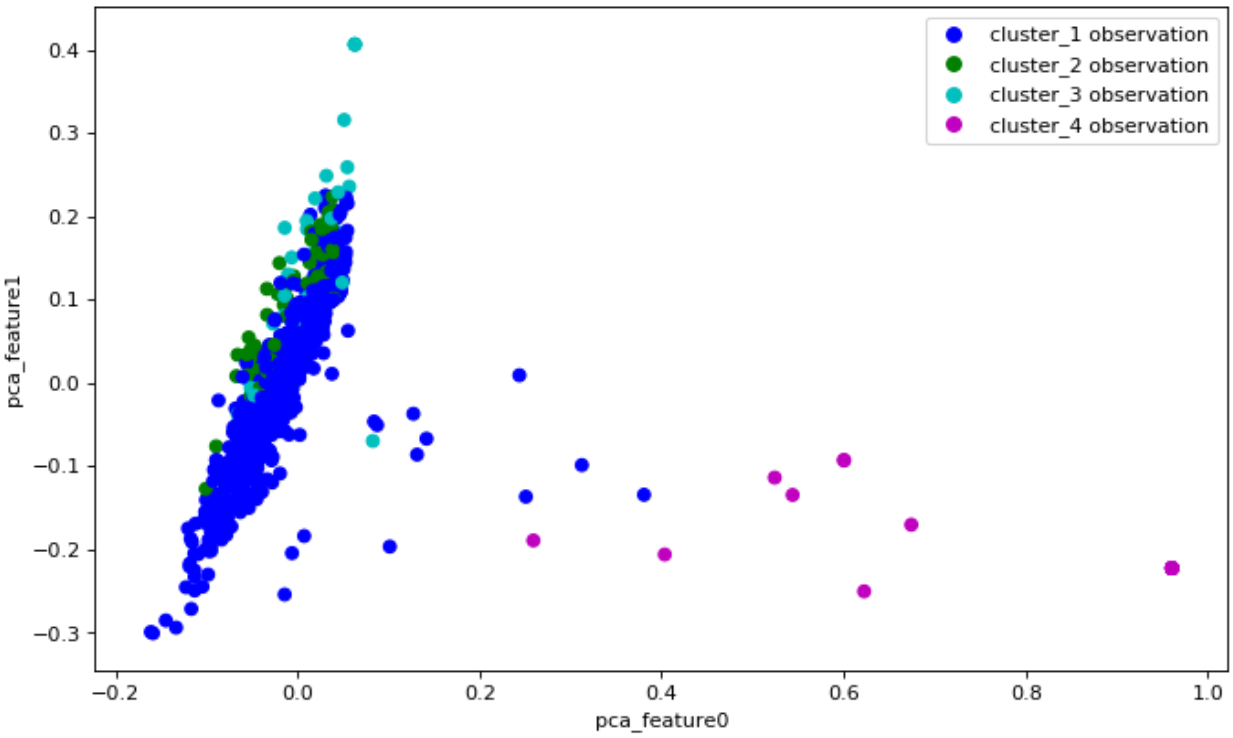
- *Agglomerative:* This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- *Divisive:* This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

In general, the merges and splits are determined in a greedy manner. In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets. [8]

Feature map dimensions = 895x3614. Number of clusters = 4.

Output cluster	Number of Observations
cluster_1	779
cluster_2	58
cluster_3	38
cluster_4	20

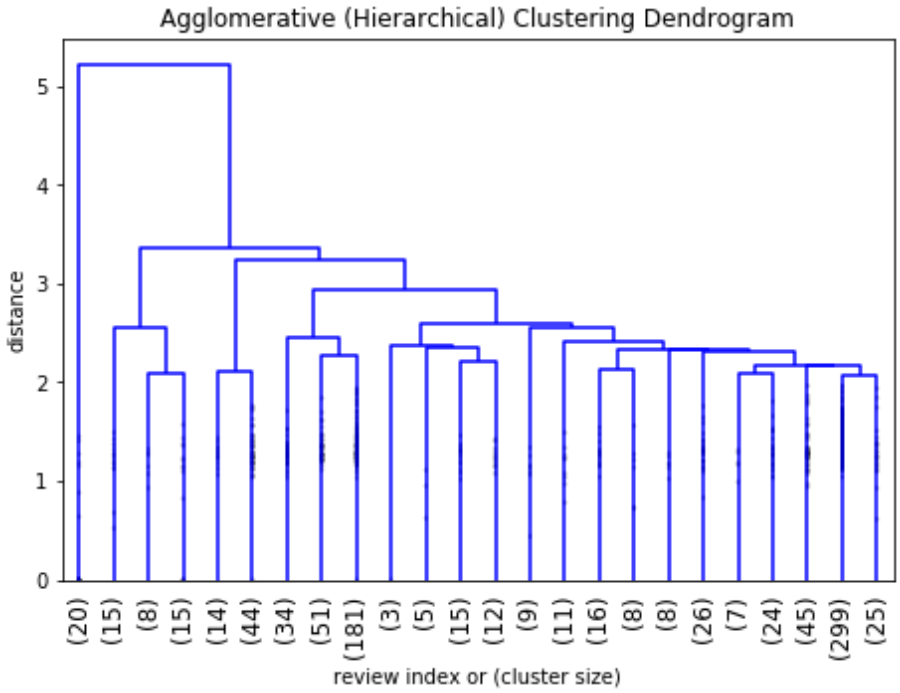
A. Using Principal Component Analysis (PCA), two representative features were identified from the tfidf feature map and the observations plotted:



B. Overall Silhouette Score = 0.017567588947823294

C. Calinski and Harabaz Score = 8.713

D. Dendrogram representation of the hierarchy tree in agglomerative clustering:



6.2 ALTERNATIVE (2) Birch Clustering

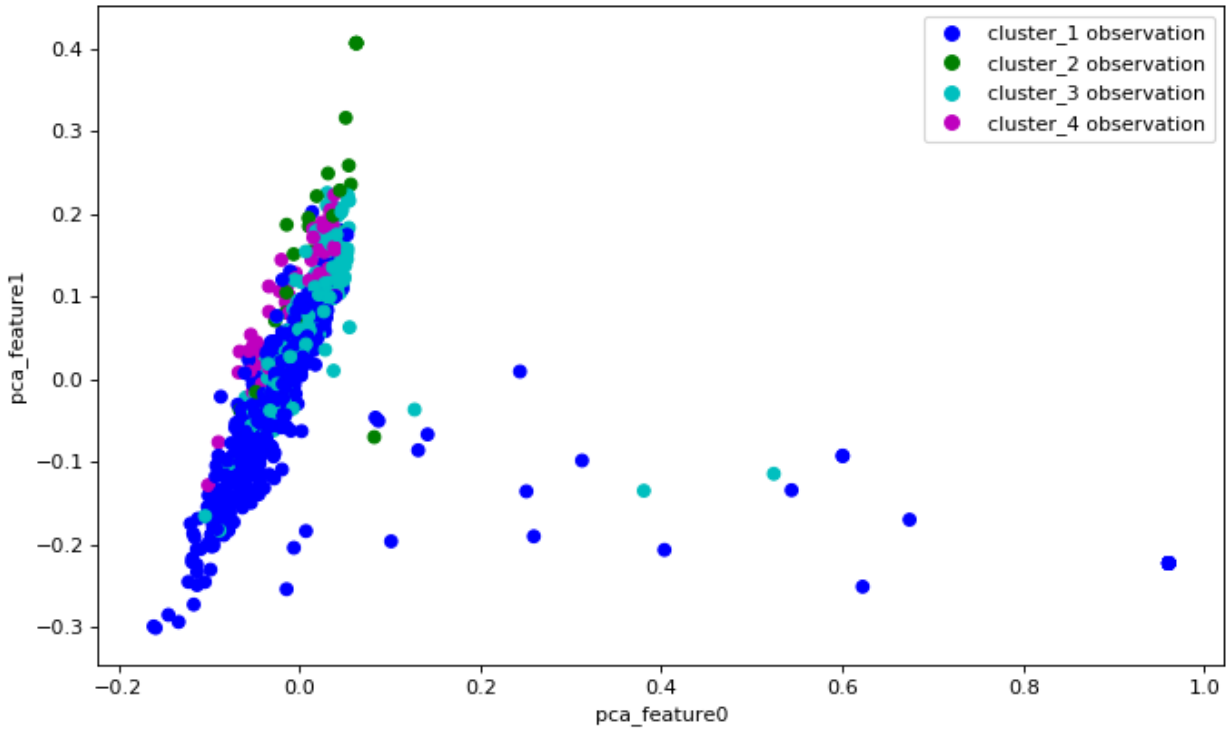
BIRCH (balanced iterative reducing and clustering using hierarchies) is an unsupervised data mining algorithm used to perform hierarchical clustering over particularly large data-sets.

The BIRCH algorithm takes as input a set of N data points, represented as real-valued vectors, and a desired number of clusters K. The first step builds a CF tree out of the data points, a height-balanced tree data structure. In step 2, the algorithm scans all the leaf entries in the initial CF tree to rebuild a smaller CF tree, while removing outliers and grouping crowded sub-clusters into larger ones. In step three an existing clustering algorithm is used to cluster all leaf entries. In step 4, the centroids of the clusters produced in step 3 are used as seeds and redistribute the data points to its closest seeds to obtain a new set of clusters. [9]

Feature map dimensions = 895x3614. Number of clusters = 4.

Output cluster	Number of Observations
cluster_1	561
cluster_2	30
cluster_3	246
cluster_4	58

A. Using Principal Component Analysis (PCA), two representative features were identified from the tfidf feature map and the observations plotted:



B. Overall Silhouette Score = 0.0071699601544512786 (very close to zero; clusters are overlapping considerably)

C. Calinski and Harabaz Score = 5.239

Conclusively,

I. SilhouetteScore(k-means) > SilhouetteScore(agglomerative) > SilhouetteScore(BIRCH), i.e., the intra-cluster similarity and inter-cluster difference are larger in k-means clustering, followed by agglomerative clustering and BIRCH.
II. Calinski-HarabazScore(k-means) > Calinski-HarabazScore (agglomerative) > Calinski-HarabazScore (BIRCH), i.e., k-means clustering performed better than agglomerative clustering and BIRCH clustering.
III. k-means clustering resulted in the least uneven cluster sizes (20, 426, 68, 381). Agglomerative clustering resulted in extremely disparate cluster sizes (779, 58, 38, 20). BIRCH was somewhere in between with (561, 30, 246, 58).

7. References

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/00461/

[2] Rajaraman, A.; Ullman, J.D. (2011). "Data Mining". Mining of Massive Datasets (PDF). pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

[3] Albitar, Shereen & Fournier, Sébastien & Espinasse, Bernard. (2014). An Effective TF/IDF-based Text-to-Text Semantic Similarity Measure for Text Classification. 10.1007/978-3-319-11749-2_8.

[4] P. Bafna, D. Pramod and A. Vaidya, "Document clustering: TF-IDF approach," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 61-66. doi: 10.1109/ICEEOT.2016.7754750.

[5] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281&ndash, 297. MR 0214227. Zbl 0214.46201.

[6] Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics. 20: 53&ndash, 65. doi:10.1016/0377-0427(87)90125-7.

[7] Liu, Yanchi & Li, Zhongmou & Xiong, Hui & Gao, Xuedong & Wu, Junjie. (2010). Understanding of Internal Clustering Validation Measures. Proceedings - IEEE International Conference on Data Mining, ICDM. 911-916. 10.1109/ICDM.2010.35.

[8] Rokach, Lior, and Oded Maimon. "Clustering methods." Data mining and knowledge discovery handbook. Springer US, 2005. 321-352.

[9] Zhang, T.; Ramakrishnan, R.; Livny, M. (1996). "BIRCH: an efficient data clustering method for very large databases". Proceedings of the 1996 ACM SIGMOD international conference on Management of data - SIGMOD '96. pp. 103–114. doi:10.1145/233269.233324.