Unsupervised Algorithms in Document Clustering

Name: Thomas Brink
ICME, Stanford University
tbrink@stanford.edu

Name: Quinn Hollister
ICME, Stanford University
bh9vw@stanford.edu

Abstract

In this report, we apply five unsupervised learning algorithms to the problem of document clustering. This problem suffers from high-dimensional, sparse and noisy data with overlapping clusters, which makes document clustering a non-trivial task. We find that three K-means-based methods have difficulty finding structure within the data and clustering observations reasonably. In contrast, we show that Ensemble K-Subspaces (EKSS) does capture information in word-document data reasonably, which it does by subspace projections and combining many weak learners. By evaluating all methods on a labeled dataset, we find that EKSS consistently outperforms the K-means type methods and is the overall superior method, reaching the highest accuracy and F_1-scores. We show that GMM also has the ability to deal with overlapping clusters, reaching the highest ARI score.

Keywords: Document clustering, Ensemble methods, Subspace projection

1 Introduction

Clustering analysis is a widely used technique to extract information from data. By separating data into groups between which observations are as dissimilar as possible and within which observations are as similar as possible, clustering analysis aims to form informative groups or 'clusters' [1]. Clustering has a plethora of use cases and, as such, is applied to a wide variety of domains, including document clustering [2]. In document clustering, the aim is to segment a collection of documents, as described by their contents (words), into informative groups. These clusters may then for instance be used to provide readers with similar documents to the ones they are reading (and are perhaps most interested in). Since document-word data is high-dimensional and extremely sparse, extracting useful and interpretable information via document clustering is no straightforward problem [3]. In this research, we explore and evaluate several different approaches to tackle this problem.

Our starting point is that of applying 'off-the-shelf' clustering algorithms. More specifically, we perform K-means clustering [4] and, in order to work with lower dimensional and more densely continuous data, apply SVD-reduced K-means [5]. Our third method is using Gaussian Mixture Models [6], which allow for more flexibility in the shape and size of clusters. We also apply two more complex methods, including a graph-based spectral clustering method called Bipartite Clustering [7] and a subspace clustering method called EKSS [8]. The former method allows us to simultaneously cluster documents and words, thus increasing interpretability of the clustering outcomes. The latter method uses an ensemble of weak learners for subspace clustering and works well in settings of highly noisy and overlapping clusters.

We start off by applying the first four techniques to a subset of unlabeled Wikipedia documents. We evaluate internal validation criteria, such as the Silhouette coefficient, and find that K-means, SVD-reduced K-means, and BP do not do a good job in separating the data into informative clusters, largely suffering from the problem of overlapping clusters. By visualizing clustering outcomes using t-SNE plots, we show that EKSS does not seem as susceptible to this problem and seems to be able to extract and separate information from noisy and overlapping clusters. To quantify our suspection, we apply all five methods to cluster a labeled dataset of news articles and report accuracies, F_1-scores, and the adjusted Rand index (ARI). Our results confirm that EKSS is the superior method for dealing with noisy and overlapping document-word data. Namely, EKSS reaches an accuracy of 48% and F_1-score of 0.25, whereas the second best method reaches values of 41% and 0.20, respectively. GMM obtains the highest ARI (0.34), which confirms that this method also has the ability to deal with overlapping clusters relatively well. Surprisingly, BP reaches the lowest accuracy, F_1-score, and ARI, barely outperforming a random clustering. Still, we show that the co-clustering of words and documents performed by BP can provide users with useful insights.

This paper will continue with a brief discussion of our data. Then, we will outline the five different methods that we apply, present our experiments and evaluation criteria, and discuss the issue of overlapping clusters. After that, we present the results from our experiments, and, last, we provide a conclusion and next steps for future research.

2 Data

Our first and main dataset is a subset of Wikipedia articles entitled 'smallwiki' [1]. This dataset includes tf-idf scores for close to 16K documents and 10.5K unique words. In addition to thus being high-dimensional, the data contains a lot of sparsity, as words typically only occur in a few documents. More specifically, only 0.5% of the data is dense, which corresponds with 675K unique word-document combinations. Because of the high level of sparsity, we process our data in CSR-format. In order to determine the rank of our tf-idf data matrix, we perform a truncated singular value decomposition and analyze the magnitude of the singular values. From Figure 1, we can see that most of the variation in the data comes from the first 75-100 singular values. This provides insight into the subspace size we should reduce our data to when applying SVD-reduced clustering.

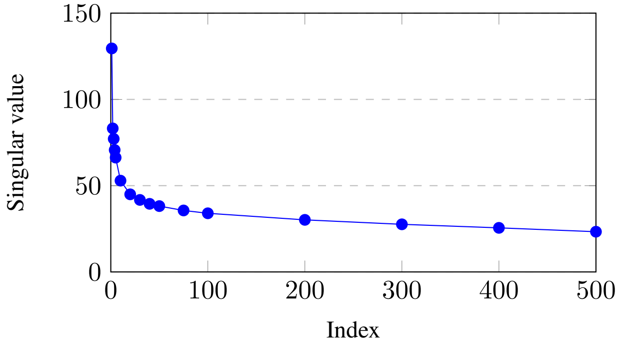


Figure 1: Magnitude of singular values versus their index for the Wikipedia dataset.

Our second dataset is the AG News Classification Dataset [2], which allows us to work with labeled data and thus compare the clustering methods using external validation criteria. This dataset consists of TF-IDF scores for around 22K words in 10K documents, which are divided into four (balanced) classes; 'world', 'sports', 'business', and 'science'. As with our main dataset, we again deal with sparse data (0.12% is dense) and use the singular values for dimension reduction (most variation comes from the first 250 singular values).

[1] https://courses.cs.washington.edu/courses/cse599c1/13wi/datasets.html

[2] https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset

3 Methodology

We first discuss the five document clustering methods we compare. Then, we discuss selecting the number of clusters, evaluation criteria, and the overlapping clusters problem. Our code, using Python 3.9.7, is available on GitHub.

3.1 Clustering Methods

K-means This method segments observations into K clusters by alternatively assigning observations to the cluster closest to them and updating the centers of those clusters. Due to its ease of use, guaranteed convergence to a local optimum, and computational efficiency, K-means is a common and popular method. However, K-means does not guarantee convergence to a global optimum and its solutions are strongly dependent on initialization. Furthermore, the sparse nature of our document data might not suit the continuity and spherical nature that is underlying for K-means. To overcome the former issue, we implement K-means by using multiple (25) K-means++ initializations and selecting the outcome leading to the lowest objective. As for the latter, we turn to our second method.

SVD-reduced K-means (SVD-K-means) The second method we use reduces the level of sparsity and high dimensionality of our data, which should make our data easier to handle and more continuous. We do so by using truncated singular value decomposition to perform dimensionality reduction, after which we run K-means on the reduced data. As with the previous method, we run this method using 25 K-means++ initializations. In addition, it should be noted that SVD-K-means requires us to select the number of components to reduce the data to, which we select by inspecting the rank of our data through a singular value plot (see Figure 1).

Gaussian Mixture Model (GMM) Our third method is a model-based clustering method that assumes our data is generated from clusters (components) that are normally distributed. When applying GMM, we use the EM-algorithm to recover the component-specific distribution parameters. To enforce structure and keep the number of parameters manageable, we assume diagonality of the component-specific covariance matrix. In spite of this choice, GMM offers more flexibility in terms of cluster shapes than K-means. In addition, GMMs allow us to compute probabilistic cluster assignments, which might help in dealing with overlapping clusters. As GMM, just like K-means, converges to local optima, we again use multiple initializations.

Bipartite Clustering (BP) and Graph Theory Connections The fourth method we apply is the bipartite clustering method of [7], which is designed to co-cluster words and documents. We can model our document-word data as a bipartite graph, where each document is a left-side node and each word represents a right-side node. Formally, we model our documents and words as part of a larger graph $\mathcal{G} = (\mathcal{V}, E)$, which has a set of vertices $\mathcal{V} = \{1, ..., |\mathcal{V}|\}$, and a set of edges that connects vertices i, j, which we denote as $E_{i,j}$. In our model, this means that each term that appears in a document can be modeled as an edge. If we want to partition the graph, i.e., cluster our nodes into separate sets, then we can evaluate this partition by looking at the corresponding 'cut' generated by this separation. If we wanted two clusters, then this would correspond to finding $\mathcal{V}_1$ and $\mathcal{V}_2$ with $\mathcal{V} = \mathcal{V}_1 \cap \mathcal{V}_2$ such that we minimize $cut(\mathcal{V}_1, \mathcal{V}_2) = \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} M_{i,j}$. Essentially, we are trying to minimize the total amount of 'contact' or edge weights between the two partitions. If we are trying to split the vertices into K partitions, then the objective function can be extended as $cut(\mathcal{V}_1, ..., \mathcal{V}_k) = \sum_{i<j} cut(\mathcal{V}_1, \mathcal{V}_2)$.

In our model, there are two sets of vertices, representing terms and documents, with edges representing terms that appear in a specific document. We will 'co-cluster' words and clusters while enforcing the constraint that there are no edges between documents or between words. As [7] shows, the second singular vectors u_2, v_2 of the term-document matrix $\mathbf{X}$ represent real approximations to the normalized min-cut problem. We use these normalized vectors to construct a smaller data matrix $\mathbf{Z}$ that represents the spectral embedding of the data onto our basis set of singular vectors, which will then enable us to partition our vertices depending on the projected data. We do this by running

K-means so we can obtain our word-document clusters. As this method uses K-means, we again face problems of finding sub-optimal and unstable clusters. We present the BP procedure in Algorithm 1.

Algorithm 1 Bipartite Clustering (BP)

```
$\mathbf{D}_1(i,i) \leftarrow \sum_j A_{ij}$
$\mathbf{D}_2(j,j) \leftarrow \sum_i A_{ij}$
$\mathbf{A}_n \leftarrow \mathbf{D}_1^{-\frac{1}{2}} \mathbf{A} \mathbf{D}_2^{-\frac{1}{2}}$
$l \leftarrow \lceil \log_2 k \rceil$
$\mathbf{U}_l, \mathbf{V}_l \leftarrow SVD(\mathbf{A}_n)$
$\mathbf{Z} \leftarrow \begin{pmatrix} \mathbf{D}_1^{-\frac{1}{2}} \mathbf{U}_{2,\ldots,l+1} \\ \mathbf{D}_2^{-\frac{1}{2}} \mathbf{V}_{2,\ldots,l+1} \end{pmatrix}$
Run K-means on the rows of $\mathbf{Z}$
```

Ensemble K-subspaces (EKSS) Our last method is an ensemble learner that combines many weak models formed via the K-subspaces (KSS) algorithm. Like BP, EKSS has no open-source implementation, so we implemented this algorithm ourselves. KSS is a variant of the K-means procedure, but instead of using centroids from which to calculate distances, it uses linear subspaces and the projections of the observations onto these lower-dimensional spaces to measure how far a point is from the 'center'. Just like K-means alternates between assigning clusters and forming new centroids, KSS alternates between assigning clusters - using the projection distance- and forming new subspaces by computing PCA on each of the subspaces. While this method is expected to solve the problem of overlapping clusters in a high-dimensional space, it suffers from sensitivity to initial conditions. In order to overcome this sensitivity and actually turn it into a strength, EKSS generates many weak learners and uses an affinity matrix to perform consensus clustering. In effect, we are minimizing the objective function

$$\min_{\mathcal{C},\mathcal{U}} \sum_{k=1}^{K} \sum_{i:x_i \in c_k} \|x_i - U_k U_k^T x_i\|_2^2 \tag{1}$$

where $\mathcal{C} = \{c_1, ..., c_K\}$ are the estimated clusters, and $\mathcal{U} = \{U_1, ..., U_K\}$ are the matching orthonormal subspace bases. The variance of the individual clusterings originates from the variance in the initialization of the subspace bases, which lie in the set of all orthonormal bases of the subspaces of dimension $\bar{d}$. In order to sample this space, also known as the Stiefel manifold of $\bar{d}$-frames in $\mathbf{R}^D$, we need to draw $\bar{d} \cdot D$ random samples from $\mathcal{N}(0,1)$, arrange them into a $\mathbb{R}^{DX\bar{d}}$ matrix $\mathbf{X}$, and normalize and orthogonalize the bases by applying the transformation $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-\frac{1}{2}}$. From these initialization bases, we can assign each observation to a cluster by finding the basis with the minimum projection distance. Once we assign all observations to clusters, we can perform PCA on each of the bases to form new ones, and, again, assign clusters. We can then iterate until either the objective ceases to decrease or, until we reach an iteration limit T. We use the first method with $T \approx 3$. Finally, we perform a final clustering on this affinity matrix via Spectral Clustering - much like the BP method - to generate our K clusters. Algorithm 2 presents the full procedure.

3.2 Selecting the Number of Clusters

In this report, we first apply the clustering methods from Section 3.1 to an unlabeled data set. Therefore, we do not know which and how many classes are represented by the documents. To select the number of clusters K, we thus resort to heuristic methods. For the K-means-type methods, i.e., K-means, SVD-reduced K-means, and BP, we plot the sum of squared errors against the number of clusters formed and look for a so-called 'elbow' in this plot to determine the most appropriate number of clusters. For the model-based clustering method GMM, we apply a model selection criterion called the Bayesian Information Criterion (BIC) [9], which is computed as

$$BIC = -2\log(L) + p\log(n), \tag{2}$$

Algorithm 2 Ensemble K-Subspaces (EKSS)

```
Input: Data $\mathbf{X} \in \mathbb{R}^{nxD}$; Number of candidate subspaces $\bar{K}$; Dimension of candidate subspaces $\bar{d}$;
    Number of base clusterings $B$; Number of KSS iterations $T$; Number of clusters $K$
Output: Clusters of $\mathbf{X}$ in $\mathcal{C} = \{c_1, ..., c_K\}$
    for $b = 1, .., B$ do
        $U_1, ..., U_K \sim Unif(Stiefel(D, \bar{d}))$
        $c_k \leftarrow \big\{x \in X \mid \forall j, \ \|U_k^T x\|_2 \le \|U_j^T x\|_2\big\}$
        for $t = 1, ..., T$ do
            $U_k \leftarrow SVD(\mathrm{c}_k, \bar{d}) \quad \forall k \in \{1, .., \bar{K}\}$
            $c_k \leftarrow \big\{x \in X \mid \forall j, \ \|U_k^T x\|_2 \le \|U_j^T x\|_2\big\}$
        end for
        $\mathcal{C}^{(b)} \leftarrow \{ \mathrm{c}_1, ..., c_{\bar{K}}\}$
    end for
$\mathbf{A}_{i,j} \leftarrow$(B)$^{-1}$ $\big|\{b : x_i, x_j$ are co-clustered in $\mathcal{C}^{(b)}\}\big| \quad \forall i, j \le N$
$\mathcal{C} \leftarrow SpectralClustering(A, K)$
```

where L denotes the maximized (complete) likelihood from a GMM with p parameters and n observations. Obviously, the likelihood and the number of parameters are (inversely) related to K. Generally, the most appropriate value of K is chosen by minimizing the BIC. As for the second (labeled) dataset, we have prior knowledge that four clusters are present. Therefore, we fix $K = 4$ for all of the described methods and use the corresponding outcomes to evaluate clustering performance.

3.3 Evaluation

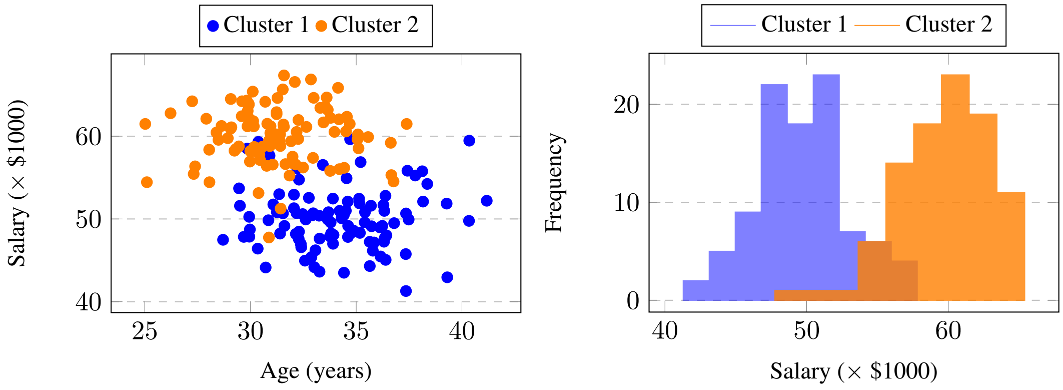
As described in Section 2, we use both unlabeled and labeled datasets. For the unlabeled dataset, we use internal validation criteria to evaluate the performance of the clustering methods. More specifically, we report the Silhouette coefficient and Davies-Bouldin index (DB), both of which use the within- and between-cluster similarity to assign a score to a clustering. For the Silhouette coefficient, a higher value indicates a better clustering, where values close to 0 indicate overlapping clusters. For DB, a lower value implies a better clustering. In addition to the above internal validation criteria, we provide some qualitative clustering insights. Namely, we use t-SNE plots to represent the clustering outcomes on a two-dimensional space.

For the labeled datasets, we can use external validation criteria, which test the clustering assignments against predefined classes. In this report, we report the accuracy, F_1-score, and Adjusted Rand Index (ARI). For all metrics, a larger value indicates a better clustering, i.e., a better correspondence between the clustering outcome and the 'true' clustering.

3.4 Overlapping Clusters

Due to the nature of our high-dimensional dataset, many clusters will have high within-cluster distance metrics, especially when we use the Euclidean metric. We can illustrate this problem with a simple example. Suppose we try to cluster two groups, based on age and salary. When we look at the entire dimension-space, we have overlapping clusters between the two groups, but there is a clear delineation when you analyze the projection onto the age subspace. Figure 2 visualizes this phenomenon. If we try and analyze this clustering without knowledge of the labels and instead just attempt to validate clustering performance by using the Silhouette coefficient or DB index, we would be tempted to disparage our results. Notice that as the dimensionality increases, the likelihood of occurrence of this phenomenon increases (which is another instance of the curse of dimensionality).

To tackle this problem, we are faced with two conclusions. First, a good way of dealing with overlapping clusters is by projecting data points onto subspaces of our ambient dimension and applying clustering to this dimension-reduced space, which our EKSS does in part. Secondly, we need to try and look for other evaluation metrics that do not completely rely on comparing intra- and inter-cluster distance on the entire space, but instead are verified through a labeled data set such as the News article dataset from Section 2.



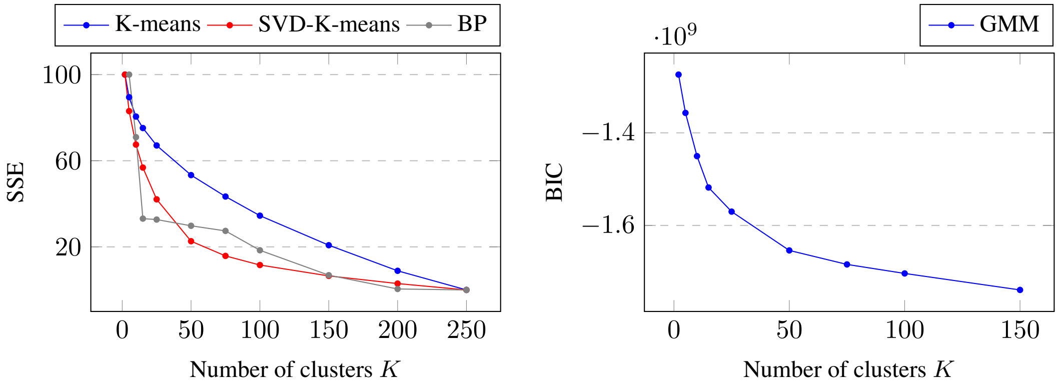
(a) Scatter plot for two overlapping clusters.

(b) Marginal histogram of salary for the two clusters.

Figure 2: Illustration of overlapping clusters and their subspace projection.

4 Results

Selecting K We start off by inspecting the number of clusters that should be selected for clustering the Wikipedia dataset. Figure 3a plots the range-normalized sum of squared errors (SSE) for K-means, SVD-K-means, and BP against the number of clusters K. A heuristic choice of K would be a point where the curve flattens out. For K-means, there does not seem to be such an 'elbow' point, although a choice such as $K = 100$ arguably provides a good choice. For SVD-K-means, the elbow point appears to lie somewhere in between $K = 50$ and $K = 100$ clusters. The BP graph flattens out already when $K = 15$, although the curve steepens again between $K = 75$ and $K = 100$. Figure 3b plots the BIC model selection criterion against the number of clusters for GMM. Theoretically, we should choose the value of K that minimizes the BIC. However, our curve seems monotonically decreasing, which is most likely due to the fact that the BIC provides an only asymptotically 'correct' selection measure. Thus, we again opt for finding an elbow, which seems to occur between $K = 50$ and $K = 100$. All in all, we decide for inspecting $K = 50$ and $K = 100$ clusters for all four methods.



(a) Normalized SSE versus number of clusters.

(b) BIC versus number of clusters.

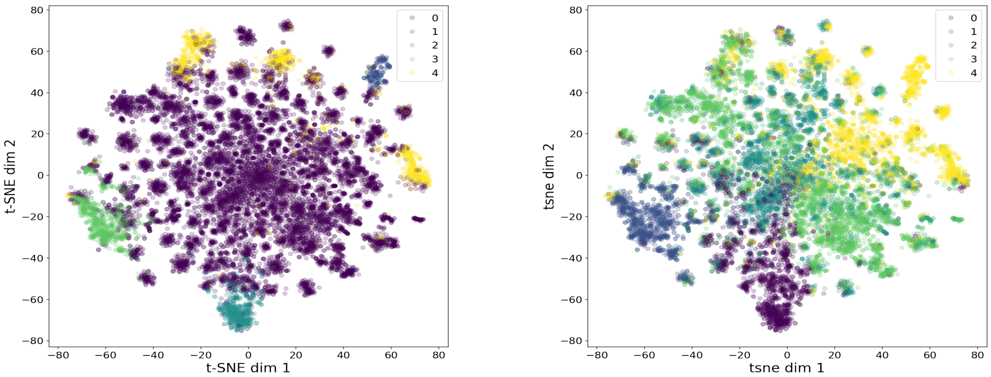
Figure 3: SSE for K-means, SVD-K-means, and BP (a) and BIC for GMM (b) versus K.

Internal Evaluation Using $K = 50$ and $K = 100$, we evaluate the performance of K-means, SVD-K-means, GMM, and BP on the Wikipedia dataset. Table 1 provides the Silhouette coefficient, Davies-Bouldin index (DB), and runtime of all four methods. In terms of DB, K-means appears to consistently perform best, while SVD-K-means and BP seem to scale best in the number of clusters in terms of runtime. As for the Silhouette coefficient, SVD-K-means seems to outperform the other two methods, though all clusterings exhibit negative coefficients, which is indicative of overlapping clusters. To solve this problem, we start looking at the EKSS method.

Table 1: *Internal evaluation measures for K-means, SVD-K-means, GMM, and BP on the Wikipedia dataset with cluster size $K = 50$ and $K = 100$.*

	K = 50			K = 100		
Method	Silhouette	DB	Time (s)	Silhouette	DB	Time (s)
K-means	-0.129	7.794	163	-0.127	6.188	262
SVD-K-means	-0.083	7.987	20	-0.113	8.230	50
GMM	-0.121	9.365	381	-0.127	7.704	675
BP	-0.202	9.007	30	-0.238	7.790	94

Clustering Visualization To provide insight into the previously described problem of overlapping clusters and the way in which EKSS handles this problem, Figure 4 provides two-dimensional t-SNE plots for K-means and EKSS using $K = 5$ clusters. It is apparent that K-means finds four small, outlying clusters and one big cluster that are all shaped near-spherically. Since the big cluster basically covers the entire dimension space (including overlap with the four smaller clusters), this method does not partition the data reasonably. In contrast, the visualization for EKSS indicates that this method finds five clusters that are more flexibly shaped, all cover different parts of the dimension space, and are less affected by outliers. In fact, the extreme groups of observations show some degree of overlap, which may be reasonable given that these are outliers. We also see some degree of overlap between cluster groups, but since we are projecting (via SVD) onto only 2 subspaces, we would not expect the overlapping cluster problem to disappear in this plot. Still, these figures illustrate the ability of EKSS to find clusters that are not just spherical, and to find multiple large groups of clusters, not just outliers. To verify that we have approximated a solution to the overlapping clusters, we would have to either visualize all of the relevant subspaces like in the toy example, or find a labeled dataset to verify our conclusions. The latter choice is much more feasible.



(a) K-means t-SNE Plot.

(b) EKSS t-SNE Plot.

Figure 4: t-SNE Plots for K-means and EKSS (K = 5).

External Evaluation Table 2 provides accuracies, F_1-scores, and ARIs for all five methods on the labeled dataset from Section 2. For the EKSS method, we assume a subspace dimension of size 5 ($d = 5$) with $T = 3$ iterations and $B = 250$ weak learners. From the table, we find that EKSS outperforms all other method in terms of accuracy and F_1-score, reaching values of around 0.47

and 0.25, respectively. In terms of ARI, GMM performs best with an ARI of 0.34, while EKSS is the second best method with a score of 0.28. This result suggests that, besides EKSS, GMM is also relatively apt to deal with overlapping clusters. However, since GMM does not perform as well in terms of accuracy and F_1, we find EKSS to be best equipped to deal with finding structure in high-dimensional and noisy datasets with overlapping clusters.

Table 2: *External evaluation measures for all five clustering methods on the News article dataset with $K = 4$ clusters. The highest score for each measure is bolded.*

	$K = 4$				
Method	K-means	SVD-K-means	GMM	BP	EKSS
Accuracy	0.4084	0.3904	0.3698	0.2577	**0.4724**
F_1	0.1895	0.1871	0.1965	0.1094	**0.2463**
ARI	0.2040	0.2127	**0.3355**	0.0000	0.2766

Interpretable Insights Although we have shown that BP obtains poor performance on our datasets, it still has an interpretability advantage in that it simultaneously clusters documents and words. These co-clusters can have many applications, e.g., clustering articles and customers of an online supermarket. Below, we provide some examples of BP clusters given by some of their top words and document titles. The first cluster corresponds with chemical documents and terms, the second cluster with (ex-)sports players and terms, and the third cluster corresponds with documents on nerves and (surgical) medical terms.

1. - Documents: *ranbp2*, *calciumphosphide*, *nectin*
   - Words: *chlorine*, *recombination*, *cations*
2. - Documents: *andreashilfiker*, *peterniemeyer*, *kevinsawyer*
   - Words: *winger*, *friendlies*, *twogame*
3. - Documents: *cuteneousnerve*, *spinalnerve*, *cutaneousnerveofforearm*
   - Words: *lymph*, *anesthesia*, *posterior*

5 Conclusion and Future Work

In this research, we evaluated five different methods for document clustering. Since word-document data generally is sparse, high-dimensional and noisy, with overlapping clusters being present, finding an appropriate document clustering method is not straightforward. By evaluating K-means, SVD-K-means, and BP on an unlabeled dataset, we find that these methods have difficulty finding structure within the data. In contrast, we illustrate that EKSS is able to find such structure by dealing with overlapping clusters through subspace projections and ensemble learning. We validate these results by applying all five methods to labeled data. EKSS is the overall superior method, obtaining the highest accuracy and F_1-score and performing relatively well in terms of ARI. We show that GMM also has some capability of dealing with overlapping clusters, obtaining the highest ARI. Surprisingly, BP performs poorest for all metrics. However, we illustrate that the co-clustering of documents and words might still be valuable to users.

For future work, we have various suggestions. First, it would be valuable to focus on computational efficiency. In this research, we were relatively limited by whether some methods could be combined with sparse matrices. Second, we would like to further investigate the selection of the number of clusters that is present in a dataset. In this research, we heuristically select this number or use prior knowledge on the ‘true’ number of clusters when selecting K, but it would be interesting to investigate or develop a method that chooses K smartly purely based on the data. Last, although we attempted to choose a diverse model group, there are other more computationally intensive methods that we would like to implement and should handle the high-dimensional problem effectively, like convex clustering, hierarchical clustering, or CLIQUE [10].

References

[1] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[2] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. 2000.

[3] Hyunjoong Kim, Han Kyul Kim, and Sungzoon Cho. Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling. *Expert Systems with Applications*, 150:113288, 2020.

[4] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[5] Khumaisa Nur'Aini, Ibtisami Najahaty, Lina Hidayati, Hendri Murfi, and Siti Nurrohmah. Combination of singular value decomposition and k-means clustering methods for topic detection on twitter. In *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 123–128. IEEE, 2015.

[6] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[7] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, 2001.

[8] John Lipor, David Hong, Yan Shuo Tan, and Laura Balzano. Subspace clustering using ensembles of k-subspaces. *Information and Inference: A Journal of the IMA*, 10(1):73–107, 2021.

[9] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[10] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1):5–33, 2005.