Project 4 – Clustering

CS548 / BCB503 / CS583 Knowledge Discovery and Data Mining - Fall 2019

Prof. Carolina Ruiz

Students: Alexander Moore and Noel Qiao

###

PLEASE NOTE PROJECT SHOULD BE 13 PAGES (13 OUT OF 13) TOTAL

###

Dataset : ● Dataset Description (not needed in this project) ● Data Exploration ● Initial Data Preprocessing (if any)	 /05 /05
Code Description:	/40
Experiments: Guiding Questions	/10
eans - Sufficient & coherent set of experiments	/10
- Objectives, Parameters, Additional Pre/Post-processing	/10
- Presentation of results	/10
- Analysis of individual experiments' results	/10
archical - Sufficient & coherent set of experiments	/10
- Objectives, Parameters, Additional Pre/Post-processing	/10
- Presentation of results	/10
- Analysis of individual experiments' results	/10
CAN - Sufficient & coherent set of experiments	/10
- Objectives, Parameters, Additional Pre/Post-processing	/10
- Presentation of results	/10

- Analysis of individual experiments' results	/10
Quantitative Analysis of Results and Discussion	/45
Qualitative Analysis of Results, Discussion, and Visualizations	/45
Advanced Topic	/30
Total Written Report Project 4	/300 = /100

Dataset Description, Exploration, and Initial Preprocessing: (at most 1 page)

[0 points] Dataset Description: (e.g., dataset domain, number of instances, number of attributes, distribution of target attribute, % missing values, ...)
(no needed in this project as this is the same dataset that you have used in previous two projects)

The Census-Income KDD Data Set is a partial collection of weighted census data from 1994 and 1995 U. S. Census population surveys. The data for this project is the training portion of the full data set, a matrix of 199523 instances and 41 attributes as it comes from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Census-Income+(KDD)). One-hot encoding encoces missing values their own column for each feature, so we use the missingness as a valid category for inferential purposes and do not unnecessarily remove data.

[05 points] Data Exploration: (e.g., comments on aspects of the dataset THAT ARE RELEVANT FOR CLUSTERING. This could include visualizations, issues with the data, and so on.)

Using summary statistics for each column after one-hot encoding gives a good impression of the data's shape by feature. The financial columns such as wage per hour, capital gains, capital losses, dividends from stocks all have extremely skewed distributions and outliers. For example, the mean of wage per hour is 55, but the max is an overflow value 9999. This will need to be addressed for clustering methods and metrics which involve distance or pairwise distance. The subset of 5% of the data (used for computational problems) left out these observations. The mean of the feature for one hot columns tells us what proportion of the rows have the feature as true, for example 47% of the observations are men (since mean(`sex`) = .47). Looking at the pairplot, we see a lot of separation in veterans.benefits splitting on age. We can see ages at and below 20 have only category 0 of veterans benefits, since people of this age cannot be veterans. Pairplot insights such as these could inform model selection decisions, and verify the quality and relationships of the data to match our assumptions. A slight positive association can be seen with age and weeks worked in a year, but most pair plots are too cluttered or high variance to gain insight on feature relationship to age. From pairplots alone, however, we can't tell much about the global relationships in the data since the dimension is so high. Data exploration findings about the extreme outliers in the numeric features is also a red flag that we will remember for experiments. High-leverage points can be harmful to naive MDS and PCA methods, and make the dimensionality reduction skewed. In addition, any other dimensionality reduction method will likely not perform well because of the high variance. The combination of high noise and some outliers means 2D visualization will be misleading. First, we must reduce the data size drastically. This is because many clustering methods, metrics, and dimensionality reduction algorithms rely on similarity or adjacency matrices, which grow as n^2 where n is the number of observations. In order to perform these methods, we first randomly subset the data down to 5% of the original size (n=199,000 to n=9,976). This lets us perform all the algorithms without computer memory issues.

[05 points] Initial data preprocessing, if any, based on data exploration findings: (e.g., removing IDs, strings, necessary dimensionality reduction, converting attributes to numeric, scaling attributes if needed, and so on.)

The following was done to both data sets as an initial preprocessing. To begin, we encoded missing values as a separate attribute whenever it appeared. We found that none of the attributes were worthy of ordinal encoding so we used one-hot encoding for every categorical attribute. We made sure that when doing the encoding, we named the new feature based on the shortest feature name with category (i.e. <Shortened Feature>: <Category>). We had to do it this way to ensure that attributes did not accidentally get overridden, and there were no repeated names. For example, for the birthplace of the observations mother and father, the same country name appears in both features. We chose to keep the full range of features made by one-hot encoding, rather than drop the kth feature (since the last feature made in one-hot encoding is redundant). This serves model coefficient and split interpretability for inference, as well as general data summary statistics even without domain or survey knowledge. These steps are the minimum necessary changes to have data ready for the clustering and visualization methods of this project.

Code Description: Python Libraries and Functions you used and what parameters you experimented with. (At most 1.75 page.)

[05 points] Preprocessing Techniques for Clustering:

We use two different data sets, since some methods and evaluations require supervision to measure cluster performance. In addition, all data columns are rescaled using MinMaxScaler to [0,1]. We also used train_test_split to sample the data down to 5% of the original data since the full data is too big to run on our computers. This issue arises when creating proximity matrices that are n by n samples large. 5% was as large as we could make it while being able to run our experiments in a reasonable amount of time.

[05 points] K-means Clustering:

We used Scikit’s KMeans method. We experiments with mainly the number of clusters to use, but also with the number of iterations until convergence. We also experimented with the maximum number of iterations to ensure that our clusters had enough time to converge.

[05 points] Hierarchical Clustering:

We used Scikit’s AgglomerativeClustering method. Again, we mainly experimented with the number of clusters, however we also used affinity to change the distance measurement and linkage to test various linkage criteria.

[05 points] DBSCAN:

We used Scikit’s sklearn.cluster.DBSCAN method for our DBSCAN experiments. DBSCAN uses two primary arguments to describe the cluster fidelity, epsilon and minimum neighbors. Since our observations are in 500 dimensional space, it takes more experiments than the ones shown to find an epsilon such that ‘neighborhoods’ are meaningful.

[10 points] Quantitative Clustering Evaluation: including metrics listed on the project description and possibly others you used

From Scikit's sklearn.metrics we used adjusted_rand_score, normalized_mutual_info_score, adjusted_mutual_info_score, and silhuoette_score for the relative indices. We also used homogeneity_score, completeness_score, v_measure_score, and contingency_matrix for our external indices. We also added a matching and similarity matrix algorithm to implement cluster correlation and the cluster correlation matrix. For K-means, we are able to use kmeans.intertia_, an attribute of the K-means class which is the SSE of the clustering. To create a proximity matrix, we used pairwise_distances from sklearn.metrics. We also wrote our own method for computing an incidence matrix.

[10 points] Qualitative Clustering Evaluation: using Visualization, including MDS and at least one more visualization technique (e.g., heatmap of the correlation between proximity matrix and incidence matrix) you used

We implemented a generalizable visualization function which used parametric MDS. We use also used sklearn.decomposition.PCA since the operation is much faster, nonparametric MDS (metric = False) is also slow but the output is even less interpretable, and heatmaps of cluster correlation. sklearn.manifold.Isomap is also used for more interpretable diagrams, though no dimension reduction technique works well on the very high dimension. We use Isomap for the best visuals to get a sense of cluster separation.

[10 points] Three Guiding Questions about the dataset domain that can be answered by Clustering methods (at most 1/4 page):

1. How many groups are typical to form in the data? What do the numbers mean?
2. How close are the points to one another? Are there noticable dense and sparse regions?
3. How does removing `sex` affect the cluster scores?

[40 points] Summary of Experiments with K-means. *At most 1 page.*

	Pre-process	**# clusters**	**Distance function**	**# iterations**	**SSE**	**% of instances per cluster**	**OTHER METRICs**	**Observations about experiment** **(e.g., observations from visualization,** **interpretation of centroids,** **analysis of similarity among instances in the same cluster)**
P31	Reduced, scaled data (unsupervised)	2	scikit Kmeans only does euclidean	n_init: 10 iterations :300	121656	50% per cluster	Adj Rand: 0 Norm Mut Info: -4 Adj Mut Info: -3.98 Silhouette: 0.334 Clus. Corr: -0.508	A 2-cluster model fails to describe the data well, and we will investigate raising K to find what meaningful partitions there are.
3P1	Reduced, scaled data (unsupervised)	10	scikit Kmeans only does euclidean	n_init: 10 iterations :300	75334	Largest 13.13%, smallest 6.68%	Adj Rand: 0.037 Norm Mut Info: 0.025 Adj Mut Info: 0.025 Silhouette: 0.338 Clus. Corr: -0.576	Significant score improvement from the 2 cluster model means that there is likely more than random association of points, and all 10 clusters are similarly sized, which means the data is likely high-noise, lacking much universal structure

P31	Reduced, scaled data (unsupervised)	20	scikit Kmeans only does euclidean	n_init: 10 iterations :300	66439	Largest 11.81% smallest 1.5%	Adj Rand: 0.077 Norm Mut Info: 0.057 Adj Mut Info: 0.055 Silhouette: 0.296 Clus. Corr: -0.549	As the number of centroids rises, the SSE will always decrease. We see clusters as small as 2% of the data, meaning there are isolated dense regions where points are contributing to high SSE (unless they are included)
31	Reduced, scaled data (supervised `sex`)	10	scikit Kmeans only does euclidean	n_init: 10 iterations :600	77534	Largest 20.5% smallest 3.38%	Adj Rand: 0.037 Norm Mut Info: 0.026 Adj Mut Info: 0.025 Silhouette: 0.338 Clus. Corr: -0.58 Homog: 0.025 Complete: 0.025 VMeasure: 0.025 ContMat: [3088, 2159], [1898, 2831]	Here, we're doubling the number of iterations in hopes that the clusters will converge more. Unfortunately for us, the clusters are already fairly close to convergence with only 300 iterations. There is no significant performance improvement.
13	Reduced, scaled data (supervised `sex`)	60	scikit Kmeans only does euclidean	n_init: 10 iterations :300	54968	Largest 8.26% smallest 0.38%	Adj Rand: 0.13 Norm Mut Info: 0.10 Adj Mut Info: 0.10 Silhouette: 0.197 Clus. Corr: -0.45 Homog: 0.09 Complete: 0.09 VMeasure: 0.09 ContMat: [3447, 1800], [1455, 3274]	By increasing the number of clusters all the way up to 60, we see that for normalized mutual information and adjusted mutual information, we get better results. However, silhouette score and cluster correlation go down indicating that our clusterings have more overlap.

[40 points] Summary of Experiments with Hierarchical Clustering (single link, complete link, average, Ward). *At most 1 page.*

1	**Pre-process**	**# clusters**	**Link type**	**Time taken**	**% of instances per cluster**	**Other Metrics**	**Observations about experiment (e.g., observations from visualization, analysis of nested clusters, analysis of similarity among instances in the same cluster)**
1	Reduced, scaled data (unsupervised)	8	average, affinity = 'cosine'	14.55 s	largest: 50% smallest: 0.02%	Adj Rand: 0.007 Norm Mut Info: 0.005 Adj Mut Info: 0.005 Silhouette: 0.27 Clus. Corr: -0.615	Using this varied parameters, we actually found comparable clustering to our 'optimal' k=8. Though, in this cluster, the largest cluster is over 50% with the smallest cluster having only a few data instances.

1	Reduced, scaled data (unsupervised)	8	complete	15.54 s	largest: 49.47% smallest: 0.09%	Adj Rand: 0.005 Norm Mut Info: 0.0025 Adj Mut Info: 0.0021 Silhouette: 0.1184 Clus. Corr: -0.574	Complete link had a similar distribution of data instances in clusters as average link however, based on metrics performed much worse.
1	Reduced, scaled data (unsupervised)	8	single	14.31 s	largest: 99.92% smallest: 0.01%	Adj Rand: 8.83*10^-5 Norm Mut Info: 0.0061 Adj Mut Info: 0.0003 Silhouette: -0.012 Clus. Corr: -0.046	Single link results, even for our 'optimal' k value ended up clustering everything into essentially one cluster with defaults for the other parameters. This performed worse than with 'ward' by a large margin.
…1	Reduced, scaled data (supervised, target = 'sex')	2	ward	15.14 s	largest: 50.44% smallest: 49.55%	Adj Rand: 0 Norm Mut Info: -4 Adj Mut Info: -3.98 Silhouette: 0.33 Clus. Corr: -0.508 Homog: 0 Complete: 1 VMeasure: 0 ContMat: [5247], [4729]	When testing one end of the spectrum of number of clusters, we found that using k=2 actually produced decent results compared to our 'optimal' value of k=8. Based only on silhouette, and cluster correlation, the results are very similar whereas for other metrics, we managed to get results that indicate an 'average random prediction.'
…1	Reduced, scaled data (supervised, target = 'sex')	10	ward	14.77 s	largest: 13.41% smallest: 2.72%	Adj Rand: 0.012 Norm Mut Info: 0.0088 Adj Mut Info: 0.0087 Silhouette: 0.331 Clus. Corr: -0.55 Homog: 0.0088 Complete: 0.0088 VMeasure: 0.0088 ContMat: [2897, 2350], [2088, 2641]	We found the optimal 'k' value to be around 8-10 with 'ward' as the link type by looking at the knees of the metrics based on k. This produced a fairly well balanced set of clusters where the smallest cluster isn't non-existent and the largest cluster is not too dominant. This result also gave us a good silhouette score and decent cluster correlation. Unfortunately for all other metrics, we got poor results.
…1	Reduced, scaled data (supervised, target = 'sex')	60	ward	15.22 s	largest: 5.96% smallest: 0.24%	Adj Rand: 0.107 Norm Mut Info: 0.08 Adj Mut Info: 0.0799 Silhouette: 0.1788 Clus. Corr: -0.394 Homog: 0.08 Complete: 0.079 VMeasure:0.08	Increasing k to 60 gave us much more balanced clusters in general and it helped boost our scores for normalized mutual information and adjusted mutual information score by a factor of 10. However, it performed much worse on silhouette indicating more overlapping of the clusters.

						ContMat: [3402, 1845], [1505, 3224]	

[20 points] Summary of Experiments with DBSCAN *At most 1 page.*

	Pre-process	**Epsilon**	**minPts**	**# clusters**	**Time taken**	**% of instances per cluster**	**Metrics**	**Observations about experiment (e.g., observations from visualization, analysis of core, border and noise points, analysis of similarity among instances in the same cluster)**
12	Reduced, scaled data (supervised)	.5	5	100 clusters, 1 outlier group	66	Smallest cluster 5 elements, largest 299, 7885 outliers	Adj Rand: 0.006 Norm Mut Info: 0.007 Adj Mut Info: 0.005 Silhouette: -.28 Clus. Corr:.14 Homog: 0.005 Complete: 0.010 VMeasure: 0.007	This and the subsequent two experiments all chose the same exactly 299 element cluster regardless of the minimum neighbors at very different thresholds.
12	Reduced, scaled data (supervised)	.5	25	13 clusters, 1 ourlier group	62	Smallest cluster 25 elements, largest 299, 8622 outliers	Adj Rand: 0.003 Norm Mut Info: 0.003 Adj Mut Info: 0.002 Silhouette: -.1 Clus. Corr:.13 Homog: 0.002 Complete: 0.004 VMeasure: 0.003	The above leads us to believe there is a truly isolated collection of 299 points in the data which could warrant further investigation

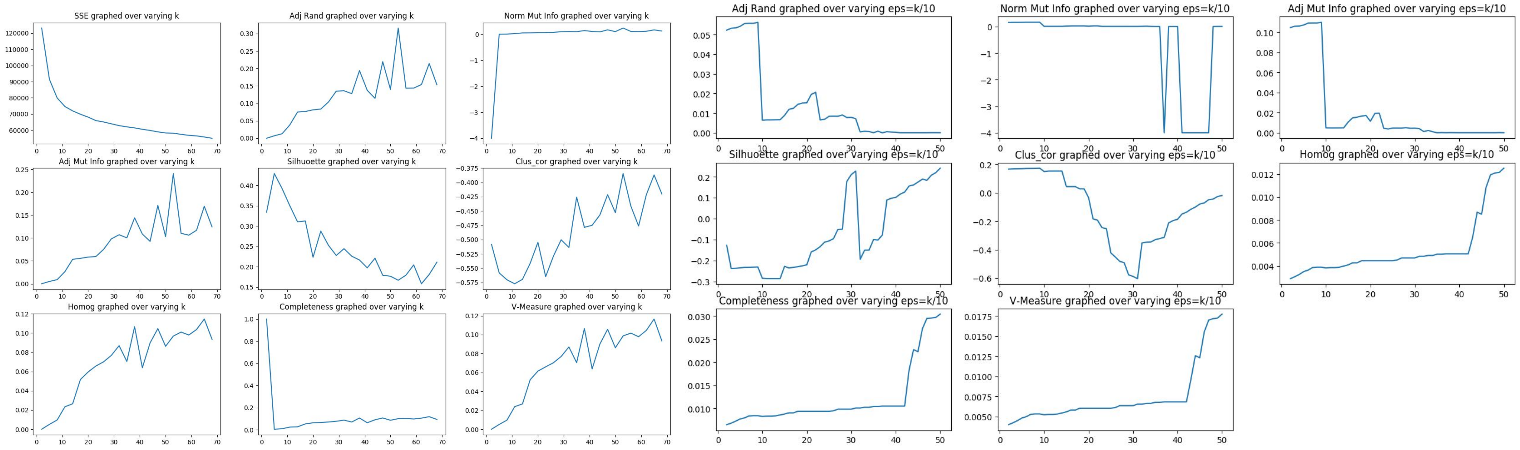
							ContMat: [254, 723]	
12	Reduced, scaled data (supervised)	.5	100	3 clusters, 1 outlier group	67	Smallest cluster 191 elements, largest 299, 9004 outliers	Adj Rand: 0.003 Norm Mut Info: 0.003 Adj Mut Info: 0.002 Silhouette: -.098 Clus. Corr:.13 Homog: 0.002 Complete: 0.004 VMeasure: 0.003	The isolated cluster could be the result of a particular type of person, or some kind of data entry problem where extrema in one variable (such as capital gains) are classed together
12	Reduced, scaled data (unsupervised)	2.3	5	36		smallest cluster 2, largest 1887, 3720 outliers	Adj Rand: 0.027 Norm Mut Info: 0.16 Adj Mut Info: 0.106 Silhouette: -.25 Clus. Corr:.17	Finding approximately ⅓ of the data to be outliers is a reasonable finding, without being able to simply inspect the high dimensional dataI chose epsilon at 2.3 because at this point we started getting a “more reasonable” mix of outliers and clustered points
12	Reduced, scaled data (unsupervised)	2.3	25	12		smallest cluster 19, largest 1128, 5522 outliers	Adj Rand: 0.027 Norm Mut Info: 0.12 Adj Mut Info: 0.075 Silhouette: -06 Clus. Corr:.15	As we raise the minimum neighbor requirement the outlier class grows, and the number of clusters goes down. This means there are large sparse regions rather than nearby dense ones
12	Reduced, scaled data (unsupervised)	2.3	100			smallest cluster 709, largest 1109, 6391 outliers	Adj Rand: 0.019 Norm Mut Info: 0.108	Requiring 100 neighbors within 2.3 distances makes the outlier class very large, where only 12 very dense clusters remain. This is likely related to the previous experiments 299 cluster.

							Adj Mut Info: 0.060 Silhouette: .20 Clus. Corr:.14	

[45 points] Quantitative Analysis of Results and Discussion (at most 2 pages).

Include here: (1) Calculations or quantitative analysis you did to obtain good parameter values for K-means and DBSCAN;

In order to find a good initial parameters for K-means, we ran all our metrics over a varying k from 2 - 50 with a step size of 3. We then graphed (on the left) all our results and found a similar knee pattern in most of the graphs which helped us narrow down the other experiments with a good starting k value. For DBSCAN, we did a similar thing where we ran all our metrics over varying episode from 2 -50 with a step size of 1 (graphs on the right). Since, k from 2 - 50 would be a huge value for eps, we are calculating eps with k/10. We found that for a given method, a good parameter is one which maximizes the silhouette score and minimizes the cluster correlation metric. For example, k=8 for Kmeans (left) and dbscan with k=30 for eps=k/10 (right) optimizes these scores for the models. We used similar methods for all models to obtain our ISOMAP visualization.



evaluation of clusterings using (2) internal indices,

We found that in general, silhouette score was maximized with less clusters and only got worse as the number of clusters increased due to more likely overlapping of clusters. As expected, SSE for Kmeans decreased as cluster numbers increased. Of course we want to maximize silhouette score and minimize SSE so finding a middle ground with the knee graph actually proved very useful.

evaluation of clusterings using (3) relative indices

For relative indices, we never got very good results for any of the metrics. We believe this is because we worked with ‘sex’ as our target and since it is a binary attribute, the clustering of data instances with 500 attributes did not necessarily separate sex well. In this case, clusterings with a majority (>50%) of ‘male’ data instances for example would cause that cluster to predict or define that clusters as ‘male’ even if the percentage of female data instances is fairly high (e.g 30% female). Though clusters don’t ‘predict’ labels, the Scikit metrics use predicted labels as a parameter, so we believe the ‘mispredictions’ we are seeing cause our scores to be very low. We did however notice that, like silhouette, the relative indices increased scores as the number of clusters increased likely due to smaller clusters which cause less ‘mispredictions’.

and (4) external indices; and

External indices rephrase the unsupervised problem into a supervised one, where we use clusters-to-labels voting to quantify the performance of different methods performed on the data. This can give us better insights into the optimal method for the domain at the cost of removing a variable (we used `sex`) from the formation of the clusters. It might be the case for methods such as DBSCAN that `sex` added a “bridge” between clusters or somehow made dense otherwise non-dense regions. Leaving out a feature with correlation to others is necessary to get meaningful external indices scores, but also likely destroys information that changes the results. Overall, external indices do give a reliable interpretation of quality if we were particularly interested in a method that maximizes one score.

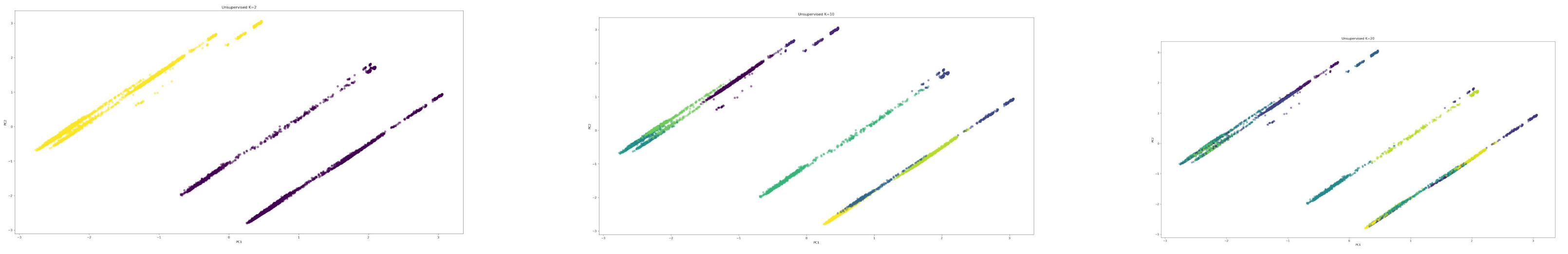
(5) other quantitative results across experiments and clustering methods. Explain your work.

When comparing k=10 for Kmeans against k=8 for Agglomerative clustering, we found that Kmeans performed much better with the relative indices. Adjusted Rand score, adjusted mutual information score, and normalized mutual information score were all considerably higher for Kmeans. This might be expected since Kmean’s interpretation of distance for centroids is damaged by the high dimension. This also might have to do with the true classes being of a more sophisticated distribution than k means, something that DBSCAN or hierarchical might be able to access but Kmeans lacks complexity.

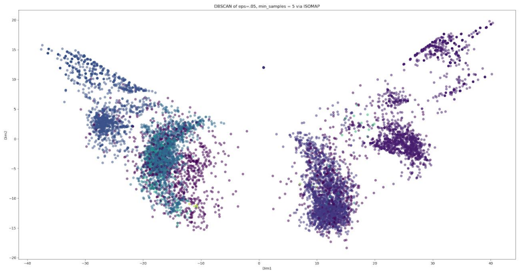
[45 points] Qualitative Analysis of Weka and Python Results on and Visualizations (at most 2 pages)

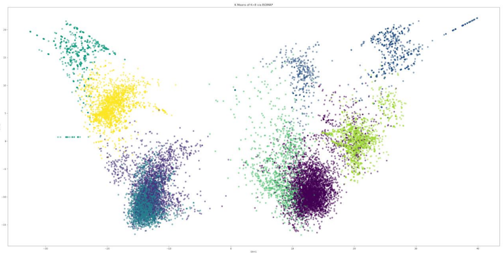
Include here (1) visualizations of the clustering using MDS and other visualization methods; (2) inspection of the actual clusters' members to find similarities among data instances in a cluster and dissimilarities with data instances in different clusters; (3) additional analysis of the results from the point of view of the dataset domain; and (4) answers that the experiments provided to your guiding questions.

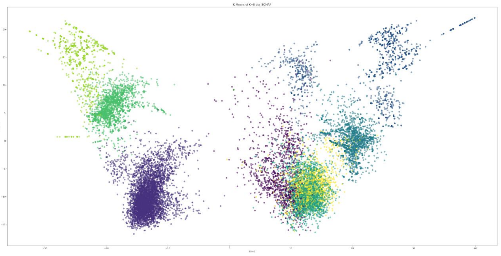
Visualization of KMeans (using euclidean MDS) according to variable K on unsupervised data



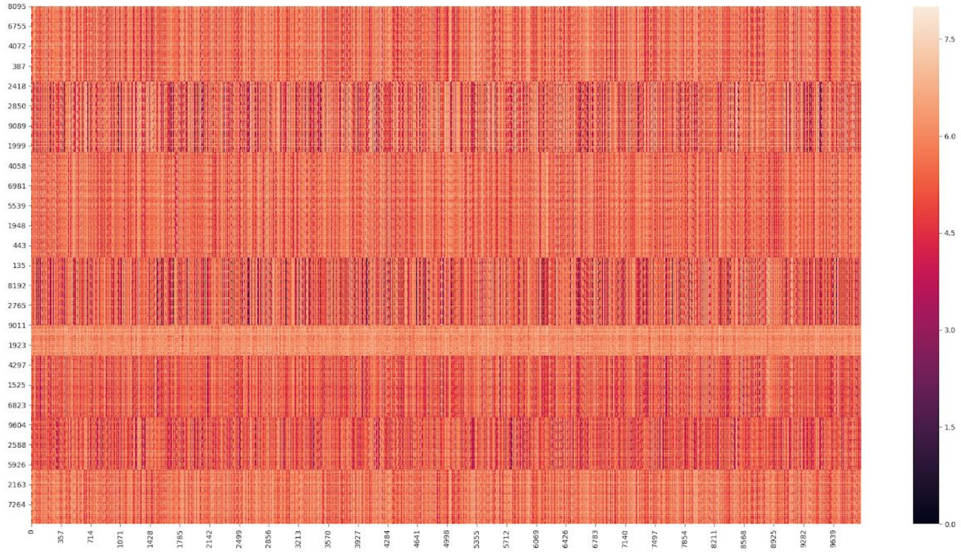
VIsualization of different algorithms via ISOMAP manifold learning







One inspection of the cluster's members to find similarities is our investigation of the 299 cluster arising from DBSCAN. Regardless of min_samples neighbors, the same 299 cluster was identified. This is a very dense region in the data we wanted to investigate. The subset has a few characteristics differing from the full data: it is entirely men with 0 wage per hour 0 capital gains, losses, stock divs, persons worked for,weeks worked, income, and unknown job type. This is a finding that we never noticed doing exploration, preprocessing, and prediction. But DBSCAN isolated these 'missing' observations because of their joint density. From the point of view of the dataset domain, these clusters will arise as a dependency of the survey structure and the way the designers encoded missingness, and the way we handled missing values. Since missing one values (such as job type) has a correlation with having other missing values, these types of rows will converge to one another and therefore create very dense regions. We can see a few such regions on the outskirts of these isomap projections (straight lines emanating outwards).



The cluster heatmap for kmeans of k=8, though sorted by cluster, shows little information encoded. This would be related to the high variability of the clustering and the noisy nature of the data.

1. How many groups are typical to form in the data? What do the numbers mean?
2. How close are the points to one another? Are there noticable dense and sparse regions?
3. How does removing `sex` affect the cluster scores?

We found resounding and shocking answers for each of our guiding questions. We found that 8 groups are typical to form in the data. They do not correspond to any particular features by inspection, but rather complex relationships which dimension reduction fails to find. We found that the majority of points are within a euclidean distance of 3 of one another (on scaled points). With an epsilon of about 2.3 we found by inspection that approximately ⅓ of the data is low-density outliers and ⅔ belongs somewhere. Removing `sex` had almost no effect on cluster scores nor cluster sizes. In a reduced dimension problem this change may have been more apparent, but the euclidean distance which algorithms rely on falls apart in high dimensions, where PNATIVITY dominates.

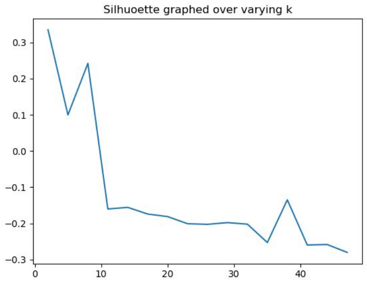
Advanced Topic: Spectral Clustering (Diffusion Maps) (at most 1 page)

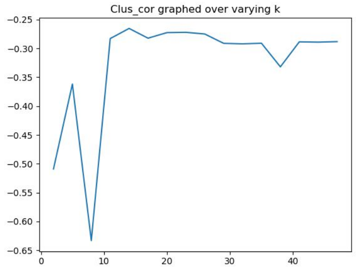
[7 points] List of sources/books/papers used for this topic (include URLs if available):

- https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf
- https://towardsdatascience.com/spectral-clustering-aba2640c0d5b
- https://calculatedcontent.com/2012/10/09/spectral-clustering/

[20 points] In your own words, provide an in-depth, yet concise, description of your chosen topic. Make sure to cover all relevant data mining aspects of your topic.

Spectral Clustering uses eigenvalues of the similarity matrix to combine dimensionality reduction and clustering approaches related to the PageRank algorithm. The use of the kernel in this method allows the algorithm to potentially get more separation in latent classes, and the benefits of reducing the dimension. Spectral clustering gives a much more robust idea of separability of clusters than a typical K-means visualized by PCA for this reason. Spectral clustering derives a Laplacian matrix from an adjacency matrix and degree matrix given by a graph representation of the data. This graph representation is learned in the kernel space with learned linked edges. The minimum-cut property of spectral clustering encodes the analog to clustering across multiple K, where the user can choose the number of minimally-invasive cuts to separate the data. We implemented Spectral Clustering using Sklearn and used our robust method evaluation and visualization tools to find optimal scores. The finding that spectral clustering agrees with the other methods that 8 clusters optimizes the scores of the data is a nontrivial finding: though each of the scores has different algorithms, spectral clustering using a kernel trick to project the data, and still finding 8 clusters, is surprising. Spectral clustering is somewhat computationally intensive but gives the user a high degree of information learned via each cut, as well as a clear derivation that does not depend on multiple iterations, stochastic properties, or hyperparameters.





Note here that spectral clustering has clear optima at 8 clusters, consistent to others.

[3 points] How does this topic relate to clustering?

Spectral Clustering is a clustering method.

Authorship: Although each student on the team is expected to be involved in every aspect of the project, describe in detail here the main contributions that each of the team members made to this project. This authorship description must accurately reflect the work done by each team member, and must be approved by all of the members of the team (at most 1/3 page)

We both drafted up the method evaluation to learn optimal K across different scores, and worked separately on visualization and evaluation methods. Anyway, how much is normal for Homeowners Association fees? I'm worried my HOA is being a little bit exploitative. I mean, what are they even spending those fees on? The landscaping looks terrible and the neighborhood playground is used by like two families, tops. Maybe the families who use the playground should be the ones paying for it? Am I the crazy one for even suggesting this? Really unfair to be charging all the houses equally when there is clearly a discrepancy in benefits is all I'm saying. Whatever, I'll drop it. It just seems unfair, that's all I'm saying.