Project 4 – Clustering
CS548 / BCB503 Knowledge Discovery and Data Mining - Fall 2017
Prof. Carolina Ruiz

Students: Janvi Kothari, Umesh Nair, Mihir Sawant

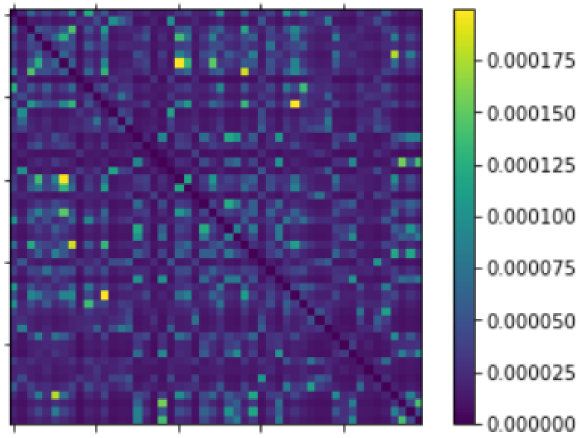
Dataset :	**Dataset**	
● Dataset Description	/05	
● Data Exploration	/10	
● Initial Data Preprocessing (if any)	/05	
Code Description: At least two Clustering algorithms	**Weka** /20	**Python** /10
Experiments: ● Guiding Questions	/10	
K-means - Sufficient & coherent set of experiments	/05	/05
- Objectives, Parameters, Additional Pre/Post-processing	/05	/05
- Presentation of results	/05	/05
- Analysis of individual experiments' results	/05	/05
Hierarchical - Sufficient & coherent set of experiments	/05	/05
- Objectives, Parameters, Additional Pre/Post-processing	/05	/05
- Presentation of results	/05	/05
- Analysis of individual experiments' results	/05	/05
DBSCAN - Sufficient & coherent set of experiments	N/A	/05
- Objectives, Parameters, Additional Pre/Post-processing	N/A	/05
- Presentation of results	N/A	/05
- Analysis of individual experiments' results	N/A	/05
Quantitative Analysis of Results and Discussion	/30	
Qualitative Analysis of Results, Discussion, and Visualizations	/30	
Advanced Topic	/30	
Total Written Report Project 4	/250 = /100	

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

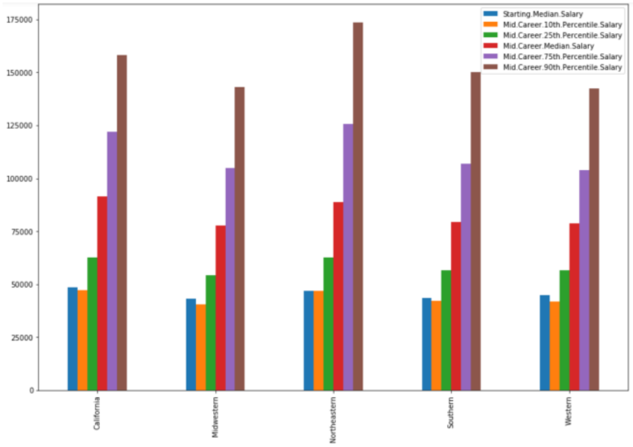
[05 points] Dataset Description: (e.g., dataset domain, number of instances, number of attributes, distribution of target attribute, % missing values, …)

The datasets used are available on Kaggle under the name “Where it Pays to Attend College”. The data was obtained from the Wall Street Journal based on data by Payscale Inc. There are three datasets, one which contains the median salaries throughout the careers of alumni of different undergraduate majors, another which contains the median salaries of alumni of different types of schools (like Party or Ivy League or State, etc.) and the last one that contains the median salaries of people that went to schools in different regions (like Northeast, Midwestern, Southern, etc.). The first dataset contains 50 instances corresponding to different majors and 8 attributes (major, starting career median salary, mid career median salary, % change from start to mid career median salary, mid career 10th, 25th, 75th and 90th percentile median salaries). The second dataset contains 269 instances corresponding to different schools and 8 attributes (school type, starting median salary, mid career 10th, 25th, 75th and 90th percentile median salaries). It has 38 NaN values which is around 14% of the dataset. The third dataset contains 320 instances corresponding to different schools and 8 attributes (region of the school, starting median salary, mid career 10th, 25th, 75th and 90th percentile median salaries). It has 47 NaN values which is around 14.68% of the dataset.

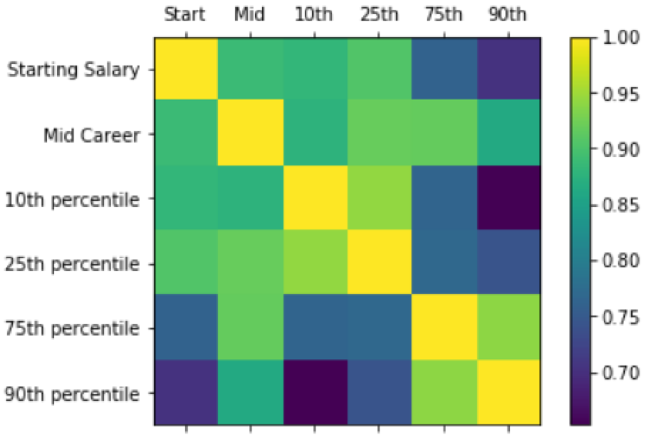
[10 points] Data Exploration: (e.g., comments on interesting or salient aspects of the dataset, visualizations, correlation, issues with the data, …)



Similarity Matrix of degrees-that-pay-back



Distribution of salaries by region



Correlation matrix of salaries on college type data

As we can see from the correlation matrix, there’s a very high correlation between the rise in salary throughout the careers.

In the similarity matrix, each cell represents the Euclidean distance between two points. Hence, the bluer the cell, the more similar the data point is and the opposite for yellow. We can thus see that most of the points are fairly similar to each other. This normally suggests that finding clusters within the data will be a little harder.

In the bar graph, we can see that northeastern schools tend to have higher median salaries than other regions, followed by Californian schools.

Weka Code Description: Inputs, output, and process followed by Weka's code for clustering (at most 2/3 page)

[10 points] Code Description of the K-means clustering algorithm implementation in Weka:

Inputs: Dataset instances, Number of clusters required, Distance function, Initialization method, Max iterations

Process: The algorithm begins by handling the missing values, if specified, with the mean (for Numeric attributes) or the mode (for Nominal attributes). Based on the initialization method, it computes as many initial centroids as there are number of clusters. When set as 'Random', it randomly selects instances from the data to be the initial centroids. Then it assigns each data instance to one of the clusters based on the centroid to which it is nearest to. The distance function used is either Euclidean(mean) or Manhattan(median) depending on the input parameter. Thereafter, the new centroid is calculated for each cluster, using the normalized values of the Numeric attributes and certain weights assigned to the Nominal attributes. The data instances are now reassigned to the clusters based on the new centroids, and the above steps are repeated till the clusters converge (centroids remain the same) or we reach Max iterations.

Output: Clustered instances

[10 points] Code Description of the hierarchical clustering algorithm in Weka: (you can pick just one "link" type: min, max, Ward's method, …)

Inputs: Dataset instances, Number of clusters required, Distance function, Link Type (say 'Complete'), DistanceIsBranchLength indicator

Process: The algorithm employs a bottom-up approach for hierarchical clustering. It begins by creating a distance matrix by computing the distance between every pair of instance in the dataset, and assigning a cluster for each instance. The distance function used is specified by the input parameter, default is Euclidean. Then it proceeds to find the best pair of clusters to merge using a priority queue based on the link type. In the Complete Link type method, the algorithm finds the largest distance between any item in one cluster and any item in the other cluster. The pair of clusters thus found are then merged into a single cluster, and the distances of the new cluster are updated in the distance matrix, along with the Priority Queue. While merging, the hierarchy of the parent and child clusters are also updated in a matrix of nodes used to keep track of the hierarchy. The DistanceIsBranchLength parameter indicates whether the distance between clusters is to be interpreted as the branch length or the node height. The algorithm further proceeds to find the next pair of clusters to merge, and the above steps are repeated till the required number of clusters are formed.

Output: Clustered instances

[10 points] Python Packages and Functions used for Clustering. Describe inputs & outputs (at most 1/3 page)

sklearn.cluster.DBSCAN sklearn.cluster.KMeans	fit(array), predict(data)	Performs DBSCAN/KMeans. Input: Parameters, Pandas Dataframe/Numpy Array. Output: Vector of assigned clusters to each data point.
sklearn.decomposition.PCA sklearn.manifold.MDS		To map the data to two dimensions for easy visualization. Input: no. of dimensions, Pandas Dataframe/Numpy Array. Output: Representation of data mapped to two dimensions.
matplotlib.pyplot		Plot visualizations. Input: Pandas Dataframe/Numpy array, various parameters to customize the plots. Output: Visualizations.
scipy.cluster.hierarchy	linkage	Performs Hierarchical. Input: Parameters, Pandas Dataframe/Numpy Array. Output: Array of clustering linkage
sklearn.metrics	silhouette_score	Evaluation metric. Input : Numpy array, parameters. Output:score as float
scipy.spatial.distance	cdist,pdist	cdist: Calculates distance between set of points. I/P:Parameters,Arrays; O/P:Array pdist: Calculates pairwise distance in n-dimensional space. I/P:Parameters,Array; O/P:Array

[10 points] Three Guiding Questions about the dataset domain (at most 1/4 page): This is for BCB503 students. CS584 questions are specified already.

[40 points] Summary of Experiments with Partitional Clustering (k-means). ***At most 1 page total, including the guiding questions above.***

	Tool	Pre-process	# clusters	Distance function	# iterations	SSE	% of instances per cluster	Observations about experiment, Observations about visualization, Interpretation of centroids, Classes to cluster evaluation?
P1	Weka	Converted Nominal to Numeric	4	Euclidean	4	50.86	Cluster 1: 20% Cluster 2: 12% Cluster 3: 26% Cluster 4: 42%	All the engineering majors are clustered together in the first cluster. The second cluster mostly consists of finance majors, while the art majors mostly make up the 3rd and 4th clusters. The 1st cluster consisted of majors with the highest salaries while the 2nd cluster had slightly lower salaries on average and so on for the 3rd and 4th clusters.
P2	Weka	Converted The salary attributes to numeric	3	Euclidean	7	274.57	Cluster 0: 45% Cluster 1: 42% Cluster 2: 13%	The clusters are analyzed wrt school type and around 53% of the instances are clustered correctly. Cluster 2 largely comprises of engineering and Ivy league schools with the highest median salaries. The other 2 clusters have similar proportions of party, lib arts and state schools and lower median salaries, according to the centroid.
P3	Weka	Converted the salary attributes to numeric	3	Euclidean	16	330.96	Cluster 0: 11% Cluster 1: 45% Cluster 2: 44%	An error rate of 67.5% was obtained. Cluster 0 mostly includes the top tech schools with the highest salaries, mostly from the NE region. Cluster 1 doesn't have any clear trends with schools from different regions from all over the country are represented. The last cluster has schools mostly from the midwestern and southern regions and they also have the lowest median salaries, according to the centroid.
P1	Python	Removed $ signs, dropped 'Undergraduate major', normalized columns	2	a. Euclidean b. Cosine c. Correlation		a. 9.6578 b. 4.6507 c. 11.933	Cluster 1: 54% Cluster 2: 46%	One cluster consists of all undergraduate Engineering majors with Computer Science majors, Economics, Accounting & Finance majors. Second cluster consists of Art History, Music, Drama, Film, Biology, History, Psychology, Sociology and majors in languages like Spanish & English. This appears to be a good clustering based on similar majors. Since the data here is normalized it is difficult to interpret the centroids. However, one cluster centroid is negative for ex. -0.1438 & other is positive i.e. 0.1688
P2	Python	Replaced missing with median of that attribute w.r.t. its School Type	7	Euclidean		148.0897	Clusters: 29.3%,27.34%,11.33%,11.33%,10.5%,7.8%,2.34%	For School Names: The centroids help to determine the salaries around which the schools are clustered together. The Ivy league colleges such as Yale, Dartmouth, Harvard are clustered together, whereas colleges famous for engineering & technology such as WPI, MIT, Carnegie Mellon are all clustered together, thus supporting our general understanding that salaries are usually based on careers domains.
P3	Python	Replaced missing values with median value of that attribute w.r.t. its Region	8	Euclidean		149.85	Clusters: 26.25%,22.18%,17.18%,16.87%,8.75%,4.37%,3.75%,0.625%	The region wise clustering does not display any evident distinguishing features. Though there are colleges such as Princeton, MIT, Harvard(Northeastern) being clustered together; they are in the same cluster with universities such as Stanford (California) and Notre Dame(Midwestern), thus making the clustering very poor based on salaries & regions.

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

	Tool	Pre-process	# clusters	Link type	# iterations	Time taken	% of instances per cluster	Observations about experiment, Observations about visualization, Classes to cluster evaluation?
H1	Weka	Dropped percent column, '$' sign, scaled down salaries	4	Single Complete Average Centroid Ward	1 for each link type	0.1s 0.07s 0.08s 0.07s 0.09s	94, 2, 2, 2 20,10,52,18 70, 26, 2, 2 70, 26, 2, 2 20,28,28,24	Single link clusters all undergrad majors, except for a few, in a single cluster. Majors related to Engineering and Math are mostly grouped in a single cluster when clustered using each of the link types. Physician Assistant and Nursing majors are placed in its own separate cluster in majority of the experiments. Interestingly, IT major was placed in a different cluster than that of Computer Science and Computer Engineering.
H2	Weka	Dropped '$' and Name column, scaled down salaries	5 3	Single Complete Centroid Average	1 for each link type	0.55s 0.45s 0.45s 0.49s	.5,1,97,1,.5 6,19,71,1,3 1,23,74,1,1 8, 91, 1	Single link clusters 97% of all the school types as 'State'. Incorrect = 32% Complete link does pretty good job of separating Engg and Ivy League types into different clusters; incorrectly clustered only 32.342% instances. Only Ward link method could cluster Party types with some accuracy. Surprisingly, the least incorrectly clustered instances (29.74%) is achieved using 3 clusters with avg link type, which clusters Engg, State and Ivy League college types more accurately than other clustering experiments. WPI repeatedly occurs alongside other top tech schools like Stanford, CMU, UCB.
H3	Weka	Dropped '$' and Name column, scaled down salaries	5	Single Complete Average Centroid Ward	1 for each link type	0.9s 0.6s 0.64s 0.59s 0.92s	1,1,0,97,1 9,18,64,8,1 1,1,26,72,1 2,1,27,69,2 10,18,40,14, 18	Single link clusters 97% of all the regions as 'Northeastern'. Incorrect = 70% Complete and Ward link distributes regions over 3 of the clusters, average and centroid link clusters regions either as Northeastern or Southern; the percentage of incorrectly clustered instances in each case lies between 67.5 to 73.44%. Centroid link type using Manhattan dist achieves lowest perc of 67.5. Only Ward link method is able to label each of the 5 clusters it generates. WPI repeatedly occurs alongside other top tech schools like Stanford, CMU, UCB.
H1	Python	Removed $ signs, dropped 'Undergraduate major', normalized columns	a. 2 b. 3 c. 2	a. Single b. Complete c. Ward		a. 0.00188s b.0.0005s c.0.00075 s	a. 4%,96% b .4%,34%, 62% c.30%,70%	The single link clustering does a good job by creating only 2 clusters, however, it just clusters Nursing & Physician Assistance into 1 cluster & the rest of undergrad majors into other. Complete link also does similar to the single link clustering where Nursing & Physician Assistant are in one cluster. The output received from Ward's is very similar to that of k-means and can be interpreted as good clustering where all science & technology majors are in one cluster.
H2	Python	Replaced missing with median of that attribute w.r.t. its School Type	2	Ward		0.004511 s	16%, 84%	The clusters obtained are just 2, based on school types or school names. This method doesn't do a good job in forming clusters using the data available. Also, visualization using dendrogram does not help us in interpreting the results.

[20 points] Summary of Experiments with DBSCAN in Python. *At most 1/2 page.*

	Pre-process	Epsilon	min Pts	# clusters	Time taken	% of instances per cluster	Observations about experiment, Observations about visualization, Interpretation of means & std dev, Classes to cluster evaluation?	Metrics (Silhouette, Homogeneity, Completeness)
D1	Remove "Major" attribute, normalize data vs original data.	a. 0.1 b. 20000	5	a. 1 b. 2	0.007 sec	For normalized: Cluster 0: 94% Outliers: 6% For original: Clusters: 70,18% Outliers: 12%	For normalized data, there are 3 outliers which correspond to the majors nursing, drama and physician assistant because of their unusual starting salary and percentage change to mid-career salary while only one main cluster was formed. In the original data, the smaller cluster contains most of the engineering majors while the bigger cluster and outlier don't have any clear trends.	S = 0.54; Other metrics are not possible since every major is unique and so there's no "ground truth".
D2	Remove school name and type, normalize data, data, replaced NA with median	a. 0.05 b. 10^4	a. 3 b.15	a. 2 b. 3	a. 0.009 sec b. 0.003 sec	For normalized: 96.65%, 1.49%, Outliers: 1.86% For original: 58.36%, 5.58%, 4.46% Outliers: 31.6%	The NMI score of the normalized data wrt to the school type as the ground truth is only 0.095 because of the dominance of one cluster. The smaller cluster consists of four lib art colleges with similar salaries. The NMI score of the original data is 0.3 which means that it has slightly "purer" clusters since NMI depends on the entropies of the classes. The higher number of outliers is because of the higher min pts threshold. WPI occurred in the bigger clusters (with no clear trends) in both results.	For normalized data: S = 0.36, NMI = 0.095 H = 0.04, C = 0.24 For original data: S = 0.13, NMI = 0.3 H = 0.29, C = 0.31
D3	Remove school name and region, normalize vs original, replaced NA with region median	a. 0.08 b. 30000	10	1	a. 0.009 sec b. 0.005 sec	For normalized: Cluster 0: 98.75% Outliers: 1.25% For original: Cluster 0: 97.19% Outliers: 2.81%	We tried running experiments over a large number of eps but there was only one interesting result. The silhouette coefficient for the clusters of the normalized data is 0.6 which implies good clustering. The 4 outliers are colleges with either very high or very low median salaries as compared to colleges similar to them. The interesting aspect of the clusters on the original data is that the outliers contain all top schools including 5 Ivy League schools and other schools with high salaries like CalTech, MIT and Stanford. WPI occurs in the bigger clusters.	For normalized data: S = 0.6, NMI =0.034 H = 0.007, C = 0.16 For original data: S = 0.62, NMI = 0.062 H = 0.018, C = 0.21

[30 points] Quantitative Analysis of Weka and Python Results and Discussion (at most 1/2 page).

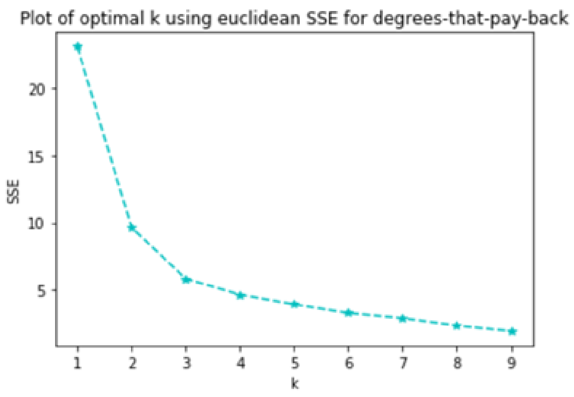
Dataset 1: K-means -> On experimenting with different k values, we obtained an optimal clustering for k=2. This can be verified by a score of Silhouette coefficient= 0.4738 (using Euclidean distance) whereas k=3 has reduced Silhouette coefficient= 0.4317 (using Euclidean dist) & = 0.4577 (using Manhattan distance). **Hierarchical** -> Majors related to Engineering and Math are mostly grouped in a single cluster, so one cluster always has more than 50% data instances. Physician Assistant and Nursing majors have lesser growth in salaries as compared to other majors, and are hence placed in a separate cluster altogether. Time taken to compute the results is less than 0.1 seconds in each of our experiments, due to the lower volume of data. **DBSCAN** -> Only 1 cluster is formed with 3 outliers of nursing, drama and physician assistant majors. However, we get a good Silhouette score of 0.54 which shows how closely the data was clustered and the presence of strong outliers.

Dataset 2: K-means -> Similarly, for this dataset we obtained elbow point at k = 7. Also obtained the Silhouette coefficient= 0.3202 (using Euclidean distance). The cluster distribution for school types is very jittered; however, there is one cluster obtained containing only Engineering school types and the rest are distributed such that Party & Liberal Arts mostly fall in the same clusters. Similarly, for school names, the IVY leagues are clustered together and Engineering colleges are clustered together. **Hierarchical** -> The lowest incorrectly clustered instances (29.74%) is achieved using 3 clusters with avg link type; With 5 clusters, Centroid Link using Manhattan distance achieves the highest recall of 86.86% in clustering State college types; Average Time taken to compute the results is 0.5 seconds. **DBSCAN** -> on the original data, it created 3 clusters. One of the clusters mostly consisted of liberal arts colleges while other majorly consisted of state schools. The evaluation metrics mostly threw up poor numbers with a Silhouette coeff of 0.13 and NMI score of 0.3

Dataset 3: K-means -> When k=3, error rate = 67.5% with an SSE of 330.96, which is higher as compared to experimenting with Python, where SSE = 149.85 using k=8. **Hierarchical** -> The percentage of incorrectly clustered instances in each case lies between 67.5 to 73.44%. Centroid link type using Manhattan dist achieves lowest incorrect perc of 67.5. Majority clustered as Northeastern; Time taken to compute the results is in the range of 0.6 to 1 second. **DBSCAN** -> The algorithm finds only one cluster on the normalized data but finds 4 very strong outliers, with a Silhouette score = 0.6. On the normalized data, again only one cluster, but outliers are mostly Ivy League colleges. The Silhouette coeff is a strong 0.62 which only reinforces how strong the outliers are.

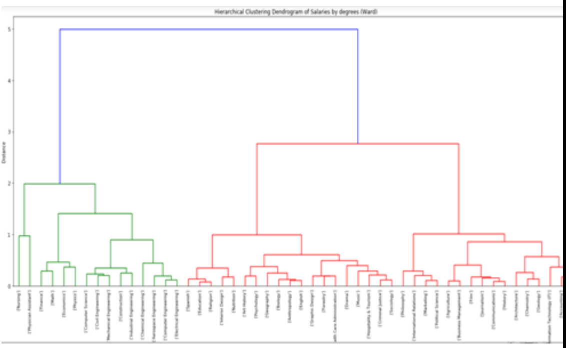
[30 points] Qualitative Analysis of Weka and Python Results on and Visualizations (at most 1 page)

K-means Clustering Analysis



On plotting SSE for different values of k in range of 1 to 10, the elbow point is obtained at k=2 for this dataset which can be seen in the plot above. Similarly, for different datasets, plotting of SSE helped us obtain k value which give out good clustering outputs.

Hierarchical Clustering Analysis



The dendrogram here shows distinguished clusters of undergraduate majors that earn varied levels of salaries. This shows that the STEM majors earn a higher salary than Social Sciences and Art majors. We have not only experienced and heard of this in our everyday lives; this is supported by the article proposed by Business Insider* which explains clearly the trend of salary distribution for different STEM & Non-STEM majors.

DBSCAN Clustering Analysis

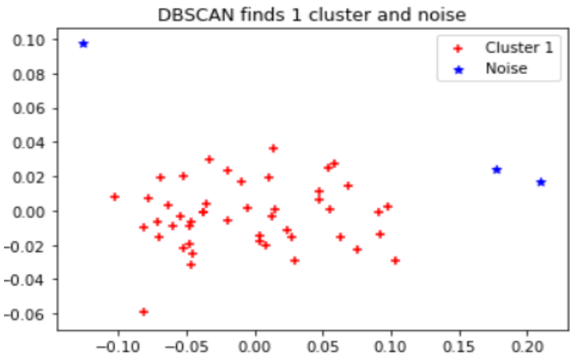
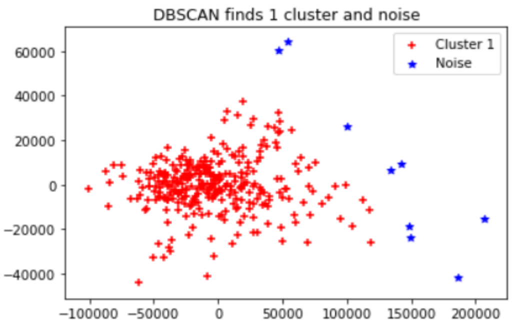


Fig. 1

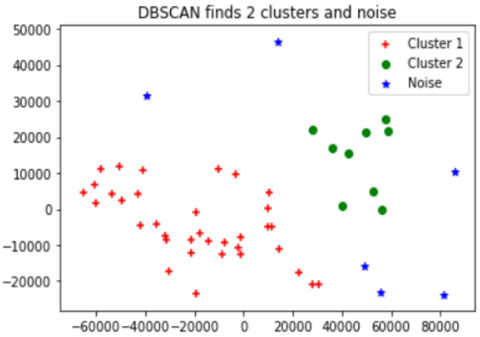
This visualization was derived from the first dataset. One of the points is a drama major who has a very low starting salary. money.usnews.com/money/blogs/my-money/2012/06/22/5-college-degrees-that-arent-worth-the-cost

Continuing on Fig. 1, another outlier is a physician assistant major who has a high salary while having an unusually very low salary curve www.forbes.com/sites/brucejapsen/2016/01/29/physician-assistant-pay-reaches-100k-annually

Nursing majors have also been counted as outliers because of their low salary curve.



This shows DBSCAN forming one cluster and a few outliers on the third dataset. All of the outliers are top tech schools which makes sense because STEM grads earn more than other graduates of other majors. www.businessinsider.com/stem-majors-earn-a-lot-more-money-after-graduation-2014-7



The above viz of dataset 1 gives two clusters. the bigger cluster doesn't possess any trends but the smaller cluster consists of mostly engineering majors that tend to pay higher. The outliers are mostly anomalies that contain majors with unusual salary trajectories over their careers.

Advanced Topic: Spectral Clustering

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

- https://www.cs.cmu.edu/~aarti/Class/10701/readings/Luxburg06_TR.pdf
- http://web.engr.oregonstate.edu/~xfern/classes/cs534/notes/Spectral-11.pdf
- http://ai.stanford.edu/~ang/papers/nips01-spectral.pdf

[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 is a technique that performs dimensionality reduction on the data before performing clustering. It uses the eigenvalues of the similarity matrix of the data, a technique used in Principal Component Analysis, a popular dimensionality reduction method. The input to this method is a similarity matrix which builds a graph structure that represents the data. The data points are represented as vertices of the graph with edges connecting them, each edge having its own weight. The weights are proportional to the similarity between the points. This graph is the depicted in the form of a matrix, whose spectrum (eigenvalues) are used to perform clustering. There are two ways to interpret Spectral Clustering, one of which is “finding partitions of the graph that minimizes normalized cut”. The objective of clustering here is to find groups within the graph that have low weights connecting one group to another and high weights connecting the data points to each other. Thus, minimizing normalized cut means minimizing the weight of connections between different groups. Another way of interpreting Spectral Clustering is with a “random walk view”. Imagine a random walk through the nodes on the graph. The objective is to find a partition of the graph such that the random walk stays within the same cluster and rarely jumps to other clusters.

The algorithm for spectral clustering is as follows:

Input: Similarity matrix of the data

- Construct a similarity graph. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L. A Laplacian is a matrix derived from a graph that can be used to find many useful properties of the graph. It “rearranges” this graph in ‘d’ dimensional space which basically makes the data easier for K-means to work on.
- Compute the first k eigenvectors of L and store these in a matrix V with the eigenvectors as columns.
- For i = 1, . . . , n, let y_i be the vector corresponding to the i-th row of V.
- Cluster the points $(y_i)_i$=1,...,n with the k-means algorithm into clusters C_1, . . . , C_k.

Output: Clusters in the data.

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

Spectral Clustering is a clustering technique, which is basically an extension to K-Means clustering, in which the data set is first mapped to a lower dimensional space and then clustering is performed. It overcomes the limitations of K-Means where if the data is ordered in certain ways, K-Means gives very poor results. Spectral Clustering uses the dimensionality of the data to overcome this.

Authorship:

Data Exploration: Each of us took up one data set and used Python for exploration, and later collaborated all our best observations.

Clustering: Each person worked on all the guiding questions related to a particular data set, and performed experiments using the different algorithms.

For the advanced topic, all of us discussed and zeroed in on this topic, then researched and documented our understanding.