Aatmay S. Talati

Machine Learning (CS 4641)

Project 3: Unsupervised Learning and Dimensionality Reduction

Date: March 31, 2018 (Spring 2018)

Analysis of Unsupervised Learning & Dimensionality Reduction

Unsupervised Learning: A Gentle Introduction

Unsupervised machine learning is the machine learning task of inferring a function to describe hidden structure from "unlabeled" data. Since the examples given to the learner are unlabeled, there is no evaluation of the accuracy of the structure that is output by the relevant algorithm—which is one way of distinguishing unsupervised learning from supervised learning and reinforcement learning [1].

In other words, unsupervised learning is a class of problems in which one seeks to determine how the data are organized. It is distinguished from supervised learning (and reinforcement learning) in that the learner is given only unlabeled examples [2].

About the Datasets:

EEG Eye State:

All data is from one continuous EEG measurement with the Emotive EEG Neuroheadset. The duration of the measurement was 117 seconds. The eye state was detected via a camera during the EEG measurement and added later manually to the file after analyzing the video frames. '1' indicates the eye-closed and '0' the eye-open state. All values are in chronological order with the first measured value at the top of the data [1].

HR Analytics:

The main goal of this dataset is to know knowing why the best and most experienced employees of the company leave the company prematurely, and also, we can predict which employee will leave the company – depending upon the inputs we

have so far. This dataset includes the reason of leaving the company/firm due to many reasons which are included in the attributes: satisfaction level, last evaluation, number of projects, average monthly salaries, time spent in company, work accidents, people left, and promotion from last 5 years, salary, and department.

Why the Datasets are Interesting?

EEG Eye State:

As a researcher working on a ground-breaking research project at Georgia Tech in the field of the Brain Computer Interface, we primarily reply on brain signals. In order to obtain brain signals we primarily use few techniques like fMRI, EEG, SSEVEP and P300.

Having this dataset has its practical use along with its real-world applicability, was the primary reason behind choosing this dataset for my machine learning project dataset. By using this dataset, we can predict what exact combination of different attributes of one EEG reading determine if the eye of the subject is open or closed and thus helping us go deeper into understanding our brain and its functioning.

HR Analytics:

In terms of the growth of the company or firm, it is very important to make sure that employees do not leave the company/firm prematurely. There are many affecting factors behind the reasons of leaving the company other than explained scenarios in the datasets, such as employer-employee relations, colleague relations, etc. This dataset has approximately 15,000 instances.

This dataset also gives some insights and makes me think that this scenario may take place one day in future, and I will know how to drill-down the main cause, like is the salary main concerned? Are promotions required for an employee to grow in his career? Is the working environment important in a company? Does the relationship between coworkers needs to be good to communicate well and stay comfortable while working? And many more.

Clustering: A Gentle Introduction

Basically, clustering is dividing and assignments into set of observations into subsets (AKA clusters). In other words, it is the method of identifying similar groups of data in a data set is called clustering. Entities in each group are comparatively more like entities of that group than those of the other groups. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields.

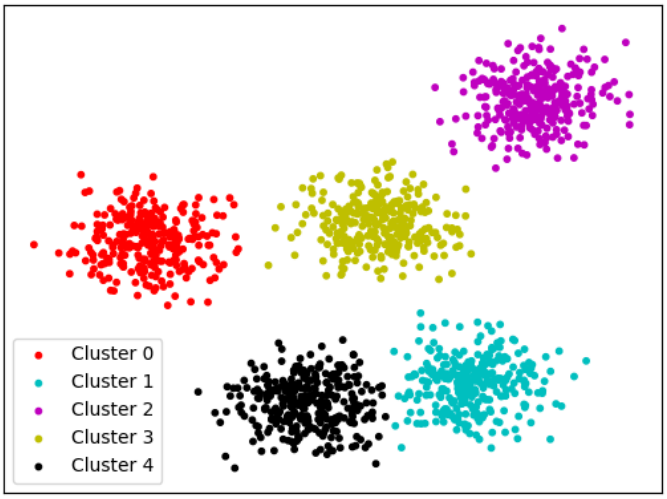


Figure 1

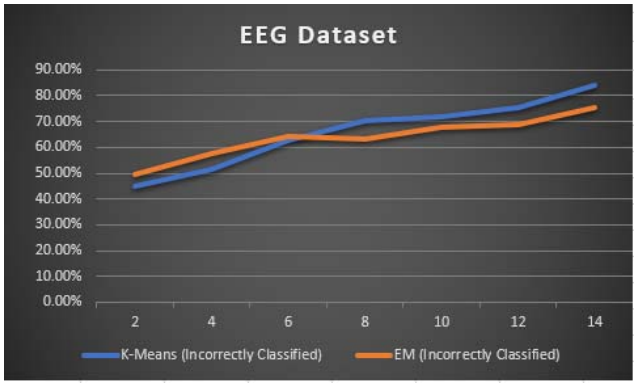
K-means clustering & Expectation Maximization (EM)

In terms of the both clustering algorithms, I've ran them on both datasets, and demonstrated the data results below. During running these two algorithms, I've varying even number of clusters starting from 2. For each number of the clusters, I've recorded the percentage of incorrectly classified clusters. The common thing for the both algorithm, in terms of the similarity between data points, I've used Euclidean distance.

EEG Dataset:

Number of Clusters	K-Means (Incorrectly Classified)	Time Taken	EM (Incorrectly Classified)	Time Taken
2	44.81%	0.26	49.62%	1.05
4	51.61%	0.27	57.40%	2.14
6	62.87%	0.83	64.03%	2.98
8	70.27%	1.69	63.39%	4.92

10	71.72%	1.24	68.01%	5.77
12	75.21%	2.43	68.75%	6.28
14	83.89%	2.01	75.40%	7.44



HR Dataset:

Number of Clusters	K-Means (Incorrectly Classified)	Time Taken	EM (Incorrectly Classified)	Time Taken
2	76.55%	0.02	76.58%	0.53
4	78.96%	0.07	75.26%	0.91
6	81.68%	0.36	77.26%	1.52
8	82.58%	0.27	77.44%	2.82
10	85.73%	0.32	81.17%	3.75
12	86.89%	0.36	80.48%	5.28
14	87.21%	0.65	82.56%	6.99

In analysis of the both datasets, we could clearly see that

$$Number\ of\ Clusters \uparrow \Rightarrow \%Incorrectly\ Classifed\ Instances \uparrow$$

And, it's pretty obvious that,

$$Number\ of\ Clusters \downarrow \Rightarrow Processing\ Time\ \downarrow$$

Overall, despite of few uncertainties before the cluster number 8 in both datasets, K-Means performed much better results than EM. We could see that in EEG dataset, EM performs nicely right after the cluster number 8, where as in HR dataset EM has many variations.

Dimensionality reduction: A Gentle Introduction

In statistics, machine learning, and information theory, dimensionality reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables. It can be divided into feature selection and feature extraction [3].

In terms of the dimensionality reduction, I've applied three algorithms on both datasets: Principle Component Analysis (PCA), Independent Component Analysis (ICA) and Randomized Optimization.

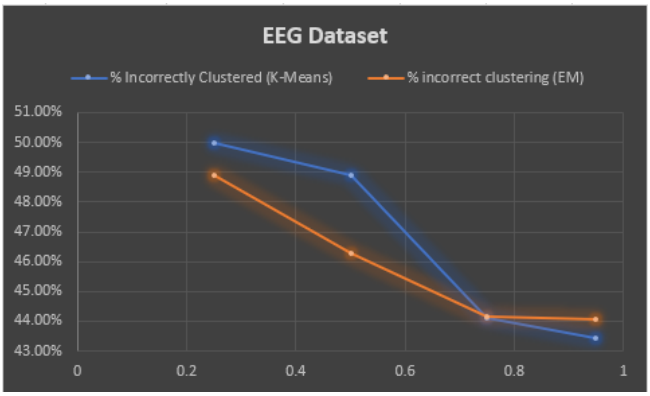
Principle Component Analysis (PCA):

In terms of the PCA, I've loaded the both datasets in Weka, and in "Preprocess" phase, I've applied filter named "principleComponent" listed under weka > filters > unsupervised > attributes. After that, I've applied K-Means and EM clustering methods on both datasets after the preprocessing phase.

While running this algorithm, I've various Variances for the same number of clusters. That has generated different number of attributes along with percentage of incorrectly clustered instances. Changes in variance has significantly influenced the number of attributes in both cases (K-Means and EM).

EEG Dataset:

Variance Covered	Number of Clusters	Number of Attributes (K-Means)	% Incorrectly Clustered (K-Means)	Number of Attributes (EM)	% incorrect clustering (EM)
0.25	2	2	49.98%	2	48.89%
0.5	2	2	48.89%	2	46.26%
0.75	2	3	44.13%	4	44.17%
0.95	2	5	43.45%	7	44.08%



HR Dataset:

Variance Covered	Number of Clusters	Number of Attributes (K-Means)	% Incorrectly Clustered (K-Means)	Number of Attributes (EM)	% incorrect clustering (EM)
0.25	2	4	50.74%	4	52.97%
0.5	2	8	50.66%	8	51.12%
0.75	2	12	46.53%	12	53.10%
0.95	2	16	46.44%	17	53.09%

As expected we could notice that,

$$Variance \downarrow \Rightarrow \ Number\ of\ Result\ Attributes \downarrow$$

Also, as we can notice one similarity in EM for EEG dataset and K-Means for HR Dataset that,

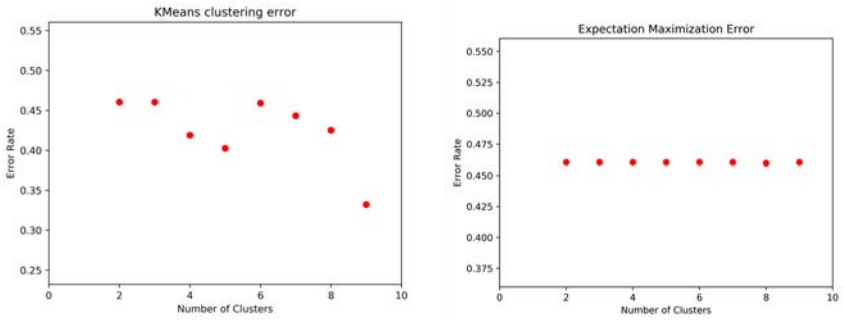
$$Variance \uparrow \Rightarrow \%\ Incorrectly\ Clustered\ Instances \downarrow$$

Also, we could see that EEG dataset worked best with PCA rather than HR. HR dataset shows many variations between K-Means and EM. That could possibly happen as most of the data in the HR dataset aren't nominal unlike EEG dataset. Reducing dimensionality would have led to an increased accuracy if the clustering algorithms, and also could have improved on time factor.

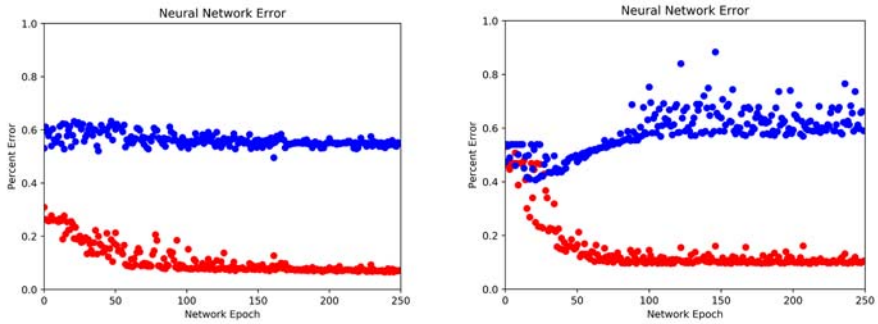
Independent Component Analysis (ICA)

After downloading and searching for FastICA filter on Weka, I had to borrow the code from my classmate, and perform ICA. It took about well over 10 hours to run on both datasets. The python code (attached in submission folder) generated following -

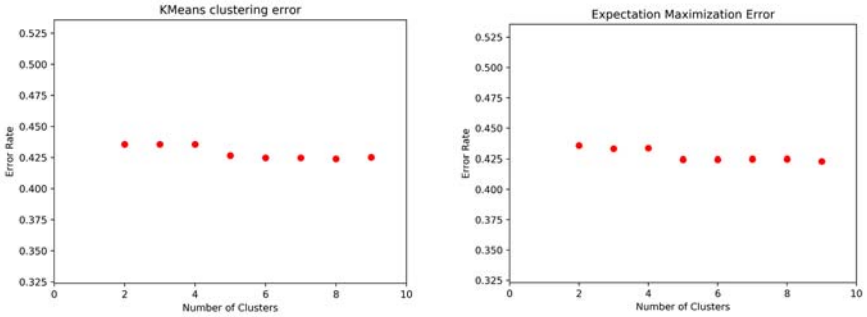
EEG Dataset:



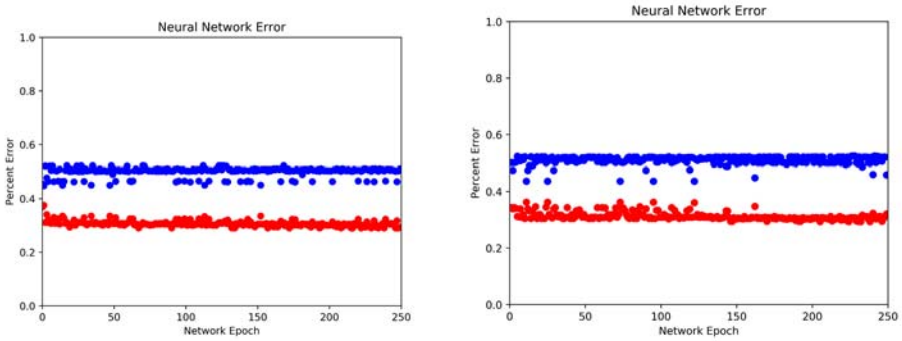
graphs as an output. Left side we see the graph for K-Means for ICA for the EEG dataset, and we can see graph of EM for ICA for EEG on right hand side. Surprisingly, we can clearly see that EM has much better results compared to K-Means.



However, we I ran Neural Network in terms of K-Means and EM for the EEG dataset, I found out that EEG dataset has error rate round about 60% on testing dataset although it performed poorly on training dataset. Whereas the EM has the error rate round about 80% on testing dataset along with poor training dataset error rate. Neural net has given the most accurate data, and I think getting the most precious result as per my expectation was having normalized data in EEG dataset.



After running on EEG dataset, as per the procedure, I ran ICA on HR dataset in terms of K-Means and EM. K-Means performed well in general, rather than EM.



On left hand side, we can see Neural net results for K-Means on HR dataset in terms of ICA, and EM on right. We can see that both algorithms worked pretty much similar in

terms of the error data in training set and testing set; however, if we look closely then we can say that K-means worked much better on ICA HR dataset rather than EM.

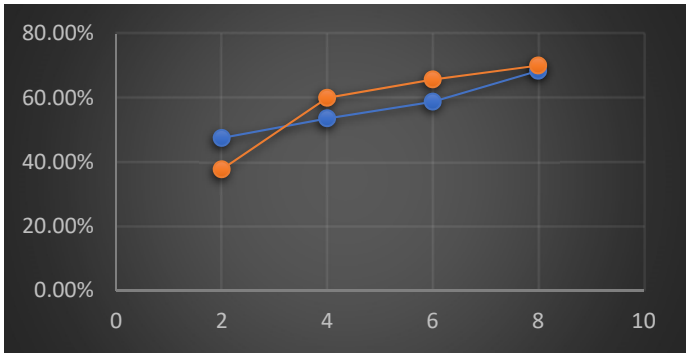
Randomized Projection

Just like we looked into PCA, in terms of the Randomized Projection, I've loaded the both datasets in Weka, and in "Preprocess" phase, I've applied filter named "RandomProjection" listed under weka > filters > unsupervised > attributes. After that, I've applied K-Means and EM clustering methods on both datasets after the preprocessing phase.

During this algorithm we have various number of clusters along with two different seed values. We will observe change in %incorrectly clustered instances along with mean squared error.

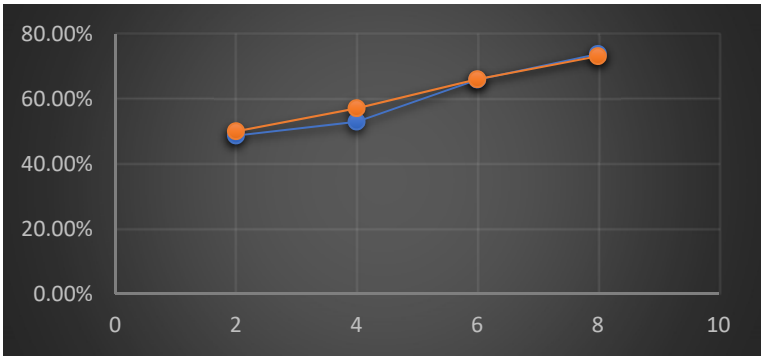
EEG Dataset

Method	Number of Clusters	%Incorrectly Clustered Instances	Mean Squared Error	Seed
K-Means	2	47.50%	13.494	42
K-Means	4	53.61%	11.7152	42
K-Means	6	58.77%	11.1508	42
K-Means	8	68.41%	1.8686	42
K-Means	2	37.73%	29.0262	51
K-Means	4	60.00%	18.6599	51
K-Means	6	65.70%	8.3651	51
K-Means	8	70.06%	5.2539	51



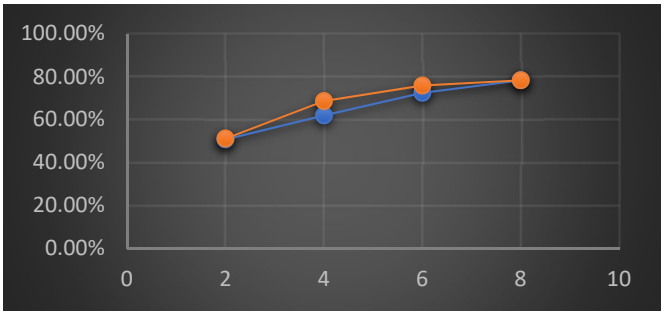
Method	Number of Clusters	%Incorrectly Clustered Instances	Log likelihood	Seed

EM	2	48.67%	-57.026	42
EM	4	52.95%	-53.9004	42
EM	6	65.90%	-53.0351	42
EM	8	73.77%	-52.2035	42
EM	2	49.98%	-57.2438	51
EM	4	57.09%	-54.5243	51
EM	6	66.01%	-53.5617	51
EM	8	73.08%	-52.9718	51

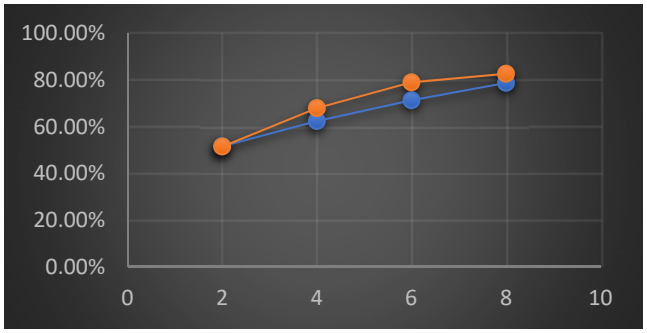


HR Dataset:

Method	Number of Clusters	% Incorrectly Clustered	Mean Squared Error	Seed
K-Means	2	50.74%	1427.9	42
K-Means	4	61.89%	1080	42
K-Means	6	72.42%	841.9	42
K-Means	8	78.26%	725.04	42
K-Means	2	51.12%	939.12	51
K-Means	4	68.61%	646.53	51
K-Means	6	75.84%	555.91	51
K-Means	8	78.26%	497.45	51



Method	Number of Clusters	%Incorrectly Clustered Instances	Log likelihood	Seed
EM	2	51.65%	-27.6	42
EM	4	62.45%	-25.54	42
EM	6	71.36%	-22.5	42
EM	8	78.80%	-21.46	42
EM	2	51.55%	-29.22	51
EM	4	67.97%	-26.95	51
EM	6	79.01%	-25.78	51
EM	8	82.72%	-25.39	51



From the given datasets, and graphs we can clearly see that,

$$Number\ of\ Clusters \uparrow \Rightarrow \%incorrectly\ classified\ instances \uparrow$$

We can also see that in terms of K-Means for both datasets,

$$Number\ of\ Clusters \uparrow \Rightarrow \%incorrectly\ classified\ instances \uparrow and\ Mean\ Squared\ Error \downarrow$$

However, in terms of the EM for both datasets.

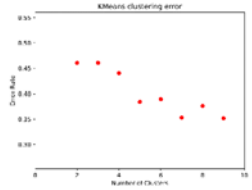
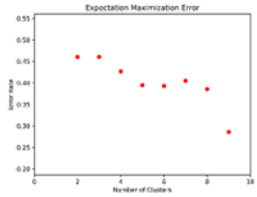
$$Number\ of\ Clusters \uparrow \Rightarrow \%incorrectly\ classified\ instances \uparrow \&\ Mean\ Squared\ Error \uparrow$$

Aforementioned various could be taking place due to the fact that random projections work better with more iterations. Increasing the seed value had a similar effect as well. K-Means for EEG dataset has few variations; other than that we can see that in rest of three scenarios, K-Means and EM closely followed each other.

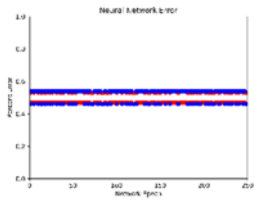
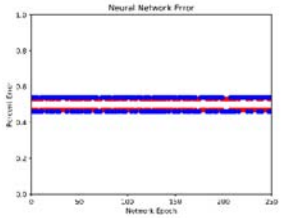
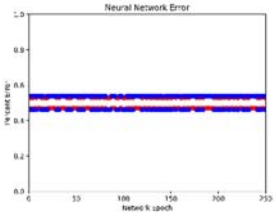
Any Feature Selection algorithm you desire : SelectKBest

The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features [4]. This algorithm nicely work in terms of selecting top k features, which has maximum number

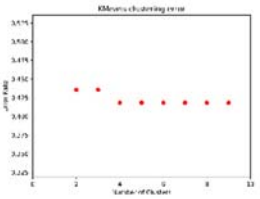
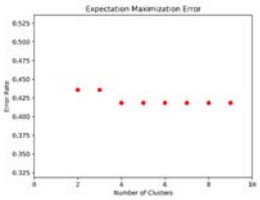
of significance with the target variable. It has two input arguments k and score_function. score_function is nothing but relevance of every feature with target variable.

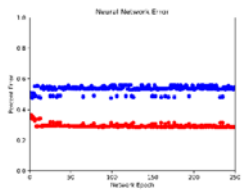
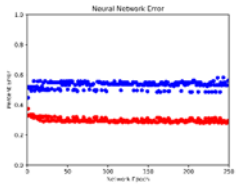
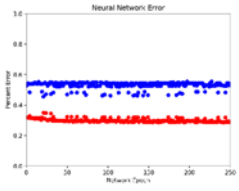
So, we ran K-Means and EM with respect to SelectKBest on EEG. We can clealy see that EM did fluctuate, but not as much as K-Means. After that I ran it thru Neural Net on the -

selectKBest algorithm and ran K-Means and EM on it. Amazingly it worked completely identical to each for testing and training datasets. After that I closely followed the same procedure on HR.

On HR dataset, K-Means and EM worked almost the same after 4th cluster. Once we applied Neural net on the SelectKBest along with HR dataset, we could see many variations

in training and testing datasets. In terms of KM and EM both had error rate close to 50%, but in the case of testing K-Means worked better, and in terms of training EM worked better. So, although we saw few variations, they all worked almost the same.

Conclusion:

Looking back, I think overall SelectKBest was one of the best choice of algorithm to look into, and implementing it. It worked with better efficiency. Despite of the SelectKBest, I think K-Means had relatively better results than EM.

References

1. https://en.wikipedia.org/wiki/Unsupervised_learning
2. http://www.cad.zju.edu.cn/home/zhx/csmath/lib/exe/fetch.php?media=2011:presentation_ml_by_ibrar.pdf
3. https://en.wikipedia.org/wiki/Dimensionality_reduction
4. https://machinelearningmastery.com/feature-selection-machine-learning-python/

Figure 1: https://www.imperva.com/blog/2017/07/clustering-and-dimensionality-reduction-understanding-the-magic-behind-machine-learning/