Randomized Optimization

Zhijian Li

1 Dataset

I will be using two datasets, both from assignment 1.
As a recap:

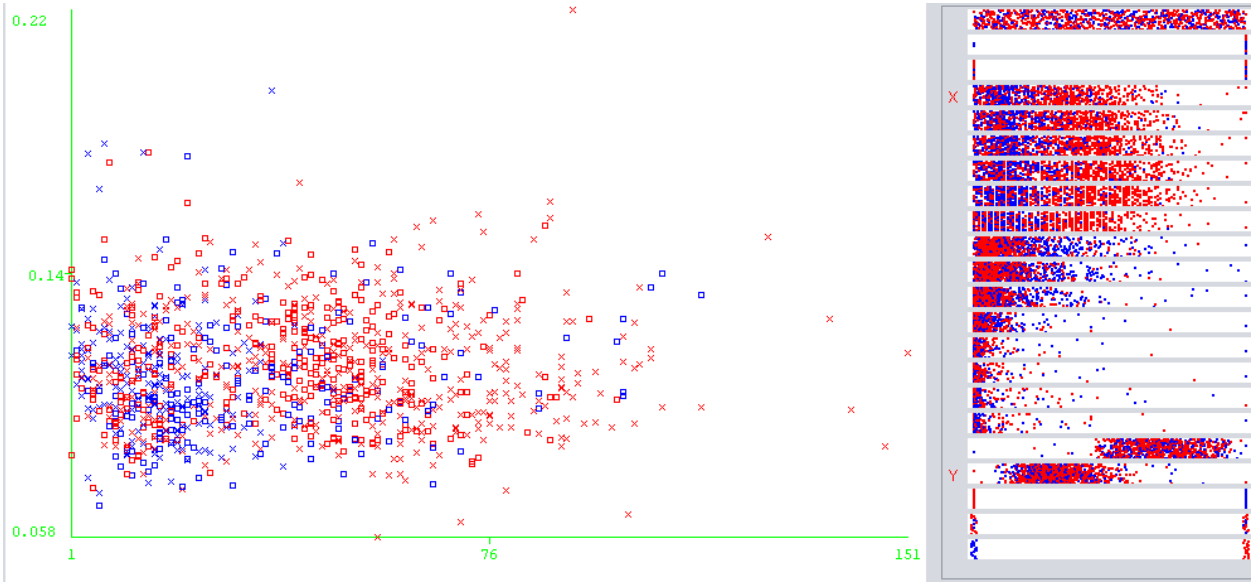
Diabetic Retinopathy Data Set (DR dataset): This dataset contains data was taken from the Messidor image set, a collection of DR examinations. The goal is to predict whether the image contains signs of diabetic retinopathy. The features are extracted from the images and is either a detected lesion, a descriptive feature of an anatomical part or an image-level descriptor. The dataset contains 1151 instances and 20 attributes.

Mammographic Mass Data Set (MM dataset): This dataset contains data from mammograms. The goal is to predict whether the patient has malignant breast cancer. The features contain BI-RADS assessments along with related measurements of the patient. The dataset contains 961 instances and 6 attributes.

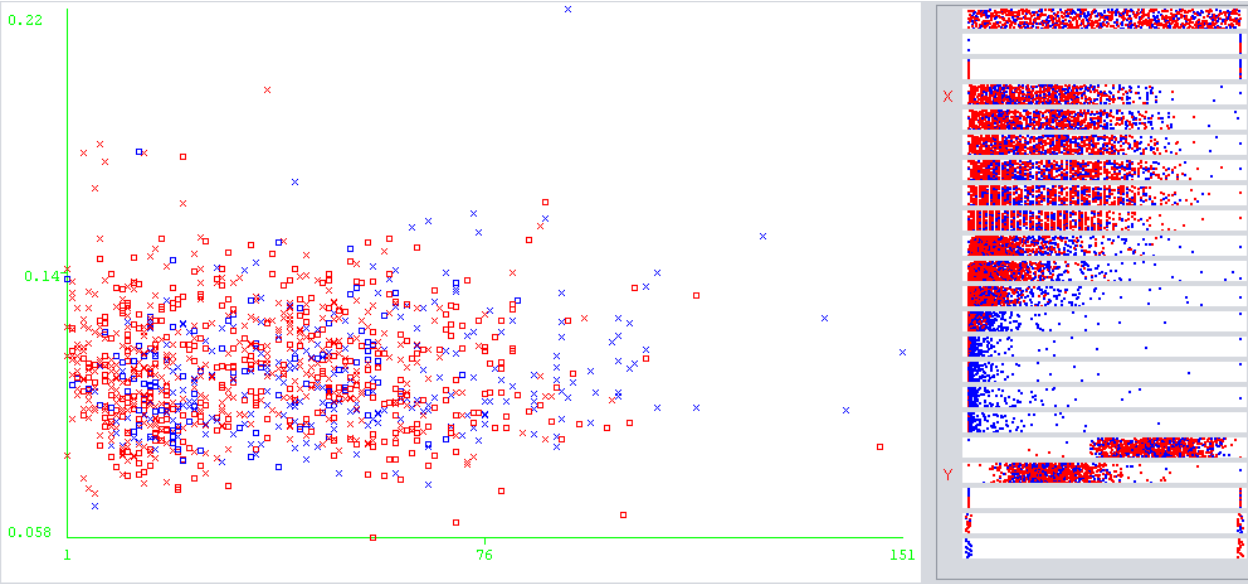
2 Clustering

I performed two clustering algorithms, K-Means and EM on both datasets. I used Weka's Classes to Cluster tool to evaluate how well each algorithm sorted the data based off the initial classification problem, severity of breast cancer (MM) and existence of diabetes (DR). I used Euclidean distance for both algorithms. Because both my datasets are binary classification problems, I set each algorithm to generate two clusters. I will be analyzing clustering between two attributes as a graph and the percent of incorrectly clustered instances.

DR



K-Means Clustering for DR



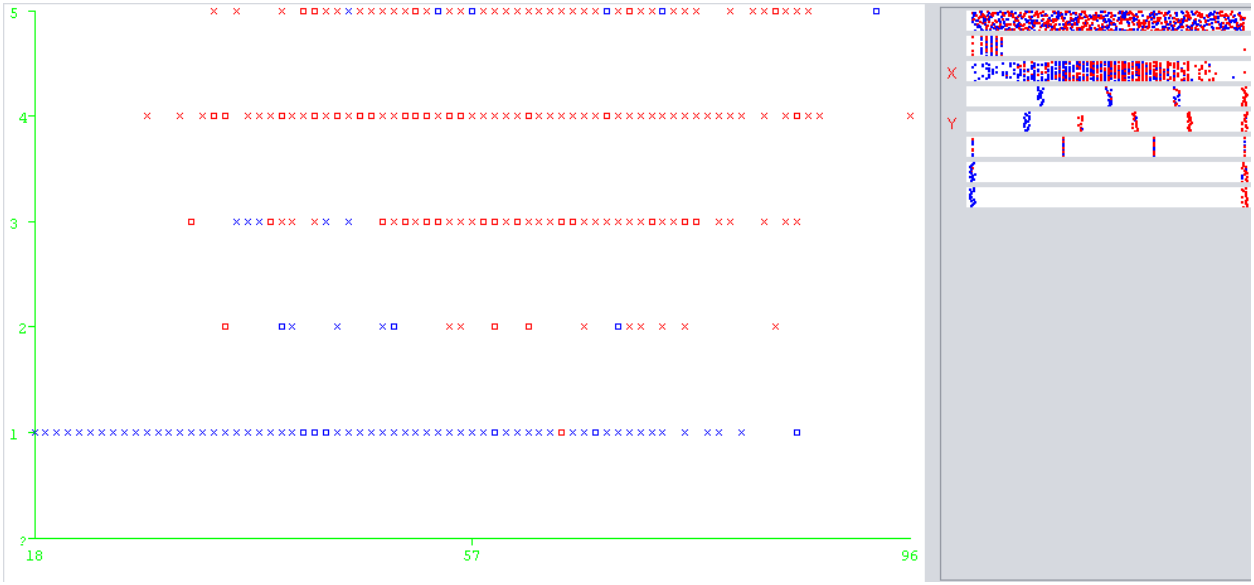
EM Clustering for DR

Algorithm	Time Taken	% Incorrectly Clustered
K-Means	0.02	47.0026
EM	0.15	41.0947

For DR, both algorithms performed relatively poorly, but EM performed better than K-Means. The graphs were taken directly from Weka, with the y-axis plotting Diameter of Optic Disc and the x-axis plotting MA Detection 1. The blue represents the cluster representing healthy patients and the red represents patients with DR, DR's dataset contained various MA Detections, from 1 to 6, all with similar graphs, so I decided to stick to MA Detection 1. I chose these two attributes because they had the best graphs, all other pairs of attributes produced graphs that were very messy, with almost no differentiation between the clusters and the axes.

As seen from the graphs, there wasn't much correlation between these two attributes and which cluster they belonged too. This was the case for every single pair of attributes and suggests that the classification problem is not linearly separable in two dimensions. This makes sense as the problem itself is a hard problem pertaining to medical diagnosis and due to the number of attributes the dataset contains. The difference is that with higher values of MA detection, K-Means classified almost always classified patients to have DR while for EM, it was more evenly split throughout.

MM



K-Means Clustering for MM

EM Clustering for MM

Algorithm	Time Taken	% Incorrectly Clustered
K-Means	0.01	21.7482
EM	0.02	26.9511

For MM, the accuracy was much better. The y-axis represents the mass margin and the x-axis represents the age of the patient. Similarly, with DR, the blue represents a benign tumor while

the red represents a malignant tumor. Despite all the data values being numerical, in reality, each number from 1 to 5 represents the different categories of mass margin (circumscribed=1 microlobulated=2 obscured=3 ill-defined=4 spiculated=5). I chose these two attributes because they highlighted the differences between the algorithms. K-Means clusters were very apparent from the graph, with benign tumors having a mass margin of mostly 1 only with no differentiation between the age of the patient. On the other hand, EM results were almost the opposite, with almost all benign tumors having a mass margins strictly greater than 1 and no differentiation between age.

Unlike the DR dataset, the MM dataset only had 6 attributes total, so there was a much higher correlation between pairs of attributes. The higher accuracy between these two datasets also suggests this. With this being the case, there was a much cleaner split between attributes and their clusters and ultimately resulted in a higher accuracy. Not only was the MM dataset a much easier classification problem, we also know now that the data was easily separable.

3 Dimensionality Reduction

Principal Component Analysis

I ran PCA on both datasets, changing the percentage of variance covered. For both datasets, PCA produced almost similar number of attributes, but this does not necessarily mean the number of attributes were reduced. For the DR dataset, the number of selected attributes decreased from 20 to 9 with 0.95 variance covered, a decrease of 11 attributes. For the MM dataset, this was not the case. While there were still 9 attributes selected, the MM dataset had 6 to begin with.

Variance Covered	# Selected Attributes
0.95	9
0.85	6
0.75	4
0.5	2

PCA on DR dataset

For the DR dataset, from the generated attributes, it is very clear which attributes from the original dataset are important and which are not. The attributes that were ranked the most important all contained linear combinations of the MA detection and MA detection exudates attributes. The ranking then went on to the diameter of the optic disc and the distance to the macula, with the quality of the assessment and results of the prescreening having much less importance. This makes sense because the most important attributes are the actual features extracted from the DR images. It is also important to note that the prescreening results have little affect for the diagnosis.

Variance Covered	# Selected Attributes
0.95	9
0.85	8
0.75	7
0.5	4

PCA on the MM dataset

For the MM dataset, there was no reduction in the number of attributes. This is because the PCA algorithm takes each of the different nominal values each attribute can take on and treats it as a separate attribute. This changed the dataset because instead of one shape and margin attribute, there were four shape attributes and 5 margin attributes, one for each value. The attributes that ranked the most important consisted of linear combinations of margin and shape. Then came age, density and assessment. This shows that the important attributes were shape and margin with the rest of the attributes having less of an effect on the outcome. This is interesting because, even with 6 attributes, PCA tells us that 2 of the attributes are very dominant indicators, while the rest of the attributes are less important.

Clustering after PCA

After PCA, the clustering algorithms showed little change for the DR dataset and showed an increase in the percent of incorrectly clustered instances for the MM dataset. For the DR dataset, this makes sense, because while PCA generates new attributes, it preserves the space between instances, thus it should not have a big impact on clustering, despite a reduction in attributes. This is not the case for the MM dataset. The MM dataset gained attributes, and this can be clearly seen in the increase in error.

Variance	Algorithm	% Incorrectly Clustered
0.95	K-Means	45.1781
0.85	K-Means	45.0912
0.75	K-Means	45.0043
0.5	K-Means	45.0043
0.95	EM	49.3484
0.85	EM	49.2615
0.75	EM	46.5682
0.5	EM	40.6603

Clustering for the DR dataset after PCA

Variance	Algorithm	% Incorrectly Clustered
0.95	K-Means	39.0129
0.85	K-Means	39.0219
0.75	K-Means	39.0219
0.5	K-Means	39.0219
0.95	EM	43.1842
0.85	EM	43.1842
0.75	EM	43.0801
0.5	EM	43.1842

Clustering for the MM dataset after PCA

DR Dataset

PCA improved K-means by about four percent while EM's performance varied. Looking at the eigenvalues, they were skewed towards the MA_Detection attributes. This is because of how PCA works. Essentially, PCA not only reduced the number of attributes, it essentially gave each

attribute a stronger or weaker rank depending on how important it was. This affects the Euclidean distance between data points because it essentially scaled all the distances by importance, so outliers have less of an impact.

MM Dataset

There was a drastic change before and after PCA. The eigenvalues were more evenly spread out, except for the last two attributes, density and assessment. Both K-Means and EM performed worse with PCA, which can be explained by the increase in the number of attributes. This can also be because of how PCA creates new attributes that are linear combinations of old attributes, increasing the complexity and making it harder to cluster.

Independent Component Analysis

I ran ICA from an open source package (student filters) in Weka. I could not tweak the number of attributes for ICA to produce, so ICA produced an equal number of attributes as the original. While it did not decrease the number of attributes, the performance of both clustering algorithms did change.

Algorithm	% Incorrectly Clustered
K-Means	46.6551
EM	42.7454

Clustering for the DR dataset after ICA

Algorithm	% Incorrectly Clustered
K-Means	27.3673
EM	24.0375

Clustering for the MM dataset after ICA

DR dataset

The results were interesting, K-Means performed better with ICA than without (46.7% vs 47%) while EM performed worse (42.7% vs 41.1%). Because the results for K-Means before and after were so similar, I can deduce that ICA did not change the distances between attributes. On the other hand, ICA seemed to have changed the probabilities for EM.

MM dataset

The results for this dataset were the opposite for the DR dataset. K-Means performed worse with ICA (27.4% vs 21.7%) while EM performed better (24.0% vs 27.0%). For this dataset, ICA modified the distances a lot, most likely because it weighted irrelevant information less (Density). For EM, this negatively affected it as it changed the probabilities.

Random Projection

I ran Random Projection on both datasets, setting the number of attributes to be 10 for the DR dataset and 5 for the MM dataset. I varied the different distributions to randomize from, from sparse1, sparse2 and Gaussian. For both datasets, I ran RP multiple times and averaged the results.

Variance	Algorithm	% Incorrectly Clustered
Sparse1	K-Means	44.4831
Sparse2	K-Means	48.2189
Gaussian	K-Means	43.7011
Sparse1	EM	42.311
Sparse2	EM	49.4353
Gaussian	EM	46.1338

Clustering for the DR dataset after Random Projection

Variance	Algorithm	% Incorrectly Clustered
Sparse1	K-Means	38.2934
Sparse2	K-Means	34.3392
Gaussian	K-Means	34.5473
Sparse1	EM	39.8543
Sparse2	EM	33.6108
Gaussian	EM	33.6108

Clustering for the MM dataset after Random Projection

After running RP, I projected the datasets back onto the original space and analyzed the distributions of attributes. The kurtosis remained the same, the visual shape of the attributes remained similar, but the mean was shifted and the variance decreased; the datasets were squished.

DR Dataset

For K-Means, it performed the best with the Gaussian distribution of random projection. There was a 4 percent decrease in error after random projection with Gaussian compared to no dimensional reduction. EM performed the best was a sparse1 distribution of random projection. Unlike K-Means, EM performed worse than before dimensional reduction.

MM Dataset

For K-Means and EM, the performance was identical for Sparse2 and Gaussian distributions. The number of incorrectly clustered points was still much worse than without dimensionality reduction.

Information Gain

With information gain, I first ran the algorithm on both datasets, without limiting the number of attributes. After that, a ranking for each attribute was given and I decided the number of attribute to keep based off this ranking. For the DR dataset, the attributes MA_Detection and MA_Detection_Exudates were ranked much higher than any of the other attributes, and the best

cutoff was 11 attributes. For the MM dataset, all the attributes had a relatively high info gain except the density attribute. So, the cutoff was 4 attributes.

Algorithm	% Incorrectly Clustered
K-Means	44.3093
EM	40.9209

Clustering for the DR dataset after InfoGain

Algorithm	% Incorrectly Clustered
K-Means	21.7482
EM	19.563

Clustering for the MM dataset after InfoGain

DR dataset

Like the other algorithms, InfoGain did not change the performance of both clustering algorithms too much, but there was an improvement to both algorithms. Because info gain ranks each attribute by the amount of information they tell you about the end diagnosis, having the clustering algorithms not only perform just as good but better than before dimensionality reduction suggests that the removed attributes were irrelevant. By removing these attributes, we essentially reduced the amount of noise in the data, allowing clustering to be more accurate.

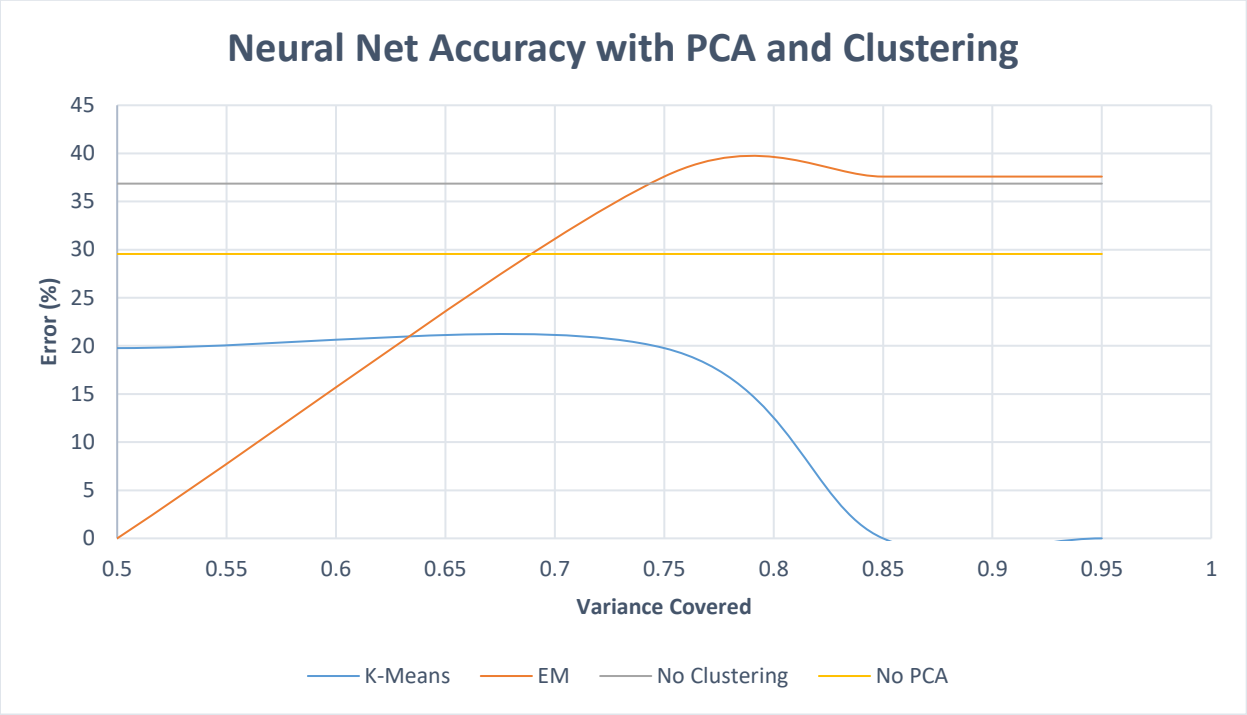
MM dataset

K-Means and EM performed surprisingly well. One simply explanation could be that infogain gave us the lease number of attributes so far for any dimensionality reducing algorithm. Both algorithms performed better than before dimensionality reduction, with EM improving almost 6 percent. This suggests that the density attribute was akin to a lot of noise, and threw off EM clustering.

4 Neural Net Performance on Altered Data

I will be running NN on DR dataset before and after both clustering and dimensionality reduction. I chose to use the DR dataset not only because it had more attributes but because it was the most consistent across clustering algorithms before and after dimensionality reduction. I was interested to see what the effects of clustering and dimensionality reduction had on the performance of neural nets. From hyperparameter tuning from assignment one, the parameters for NN I will be using is a 0.3 learning rate and 0.2 momentum with 5 hidden layers. The best accuracy NN achieved with no altered data was 29.55% error.

Principal Component Analysis



The results for PCA are very interesting. Without clustering, PCA did not improve NN's performance, but increased error by almost 7%.

At low variance covered, both clustering algorithms drastically improved NN's performance. At 0.5 variance covered, K-Means improved NN's accuracy by 10% and EM eliminated NN's error completely. The fact that K-Means and EM both achieved 100% accuracy at some point is very implausible. My clustering parameters might have been the reason for this. As PCA's variance covered increased, K-Means worked better while EM started to cause more issues.

Independent Component Analysis

Algorithm	% Error
N/A	27.03
K-Means	13.76
EM	13.76

ICA improved NN's performance, with and without clustering.

Without clustering, there was a 2.5% increase in performance. With clustering, both K-Means and EM performed equally well, lowering the error by about 14%. Runtime for NN did not decrease too much, as there wasn't any elimination of attributes.

Random Projection

Variance	Algorithm	% Error
Sparse1	N/A	27.11
Sparse2	N/A	27.62
Gaussian	N/A	26.34

Sparse1	K-Means	12.79
Sparse2	K-Means	14.45
Gaussian	K-Means	14.83
Sparse1	EM	12.79
Sparse2	EM	14.45
Gaussian	EM	14.83

Without clustering, Random Projection still improved NN's performance, by about 2% on average. This can be explained by the fact that random projection cleaned up the data, removing a little noise.

With clustering, NN's performance improved by about 13% on average. Using the Sparse1 distribution, NN reduced its error by almost 17%. For both clustering algorithms, this huge decrease in performance can be because the clusters formed gave crucial information on the classification problem.

Information Gain

Algorithm	% Error
N/A	27.49
K-Means	15.27
EM	15.27

Information gain had similar results with the other algorithms. By removing attributes that gave little information, it appears that a lot of the noise in the data was removed. It seems like in assignment 1, a lot of the attributes were misleading, which can be attributed to the quality of assessment and prescreening attributes. This suggests that these two precautions to DR almost have no effect and that this disease is almost impossible to detect with one assessment. By removing these misleading attributes, NN's performance increased a lot. Adding clustering to the dataset reduced the error by more than half.

References:

1 https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass

2 https://archive.ics.uci.edu/ml/datasets/Diabetic+Retinopathy+Debrecen+Data+Set