Jeremy Kang
Jeremy.kang@gatech.edu
CS 7641: Machine Learning – Assignment 3

Datasets

The two datasets from the assignment were used again for this assignment. They were chosen because of their relatively noise-free attributes and categorical disposition. To review, the first dataset was the car evaluation data. This data was collected to rate cars either unacceptable, acceptable, good, or very good based on six different features: overall price (v-high, high, med, low), price of maintenance (v-high, high, med, low), number of doors (2, 3, 4, 5-more), person capacity (2, 4, more), size of luggage boot (small, med, big), and safety (low, med, high). There are 1728 instances with a total of 7 attributes (i.e., including the class attribute).

The second dataset is actually a subset of a 19 where samples of married women that were not (or did not have confirmation of being) pregnant at the time of data collection. The objective of the study was to predict contraceptive method choice (no use, long-term methods, short-term methods) based on 9 dimensions:

- wife's age ([numerical]),
- wife's education (1 = low, 2, = medium-low, 3 = medium-high, 4 = high),
- husband's education (1 = low, 2, = medium-low, 3 = medium-high, 4 = high),
- number of children ([numerical]),
- wife's religion (0 = Non-Islam, 1 = Islam),
- wife's employment status (0 = yes, 1 = no),
- husband's occupation (1, 2, 3, 4),
- standard-of-living index (1 = low, 2, = medium-low, 3 = medium-high, 4 = high),
- media exposure (0 = good, 1 = not-good)

Both data sets required preprocessing. The features were converted to numeric values with the starting value of 0 (e.g., for the car data, overall price was: v-high = 0, high = 1, etc.). The class labels were also numerically converted.

In addition to the raw datasets, new sets were created for each of the dimensionality reduction algorithms for 6 additional sets (i.e., car evaluation: ICA, PCA, RCA and contraceptive method choice: ICA, PCA, and RCA). Lastly, four sets additional were created for an ANN analysis corresponding to the clusters found by *k*-means. WEKA's MultiLayerPerceptron necessitates that the class labels are nominal. Thus, they were reconverted to nominal values for the ANN algorithms.

Software and Approach

WEKA's GUI Explorer and Experimenter, as well as Microsoft Excel were used to run and analyze the algorithms.

Weka's implementation allowed for *k* to be chosen for the algorithms. However, for *k* -means, the clusters were set to the corresponding number of labels (i.e., for the car evaluation data, *k* was set to 4 and for the contraceptive method choice data, k was set to

3). For expectation maximization (EM), *k* was chosen by the algorithm and varied from 2 to 4 clusters. There are some underlying issues with the way *k* is chosen and is further addressed below.

Classification Problems

1. Classifying car [CAR] acceptability (i.e., unacceptable, acceptable, good, or very good) based on the car's overall price, price of maintenance, number of doors, person capacity, size of luggage boot, and safety.

2. Classify contraceptive method choice [CMC] (no use, long-term method, short-term method) based on wife's age, wife's education, husband's education, number of children, wife's religion, wife's employment status, husband's occupation, standard-of-living index, and media exposure.

Clustering

k-Means

SimpleKMeans clusterers were chosen for the first part of the problem. Note that the Euclidean Distance measure is defaulted into SimpleKmeans in WEKA. Four clusters were chosen for the CAR data and three clusters were chosen for the CMC data. The clusters represent the class labels in their respective datasets. This method was chosen because of how the *k*-means algorithm assigns data points to be a part of one cluster over another in order to find the centroid of the cluster. Thus, it is important to explicitly state what *k* would be before for initializing the algorithm.

kMeans - CAR

Number of iterations: 33
Within cluster sum of squared errors: 1179.6615176971243
Time taken to build model (full training data) : 0.05 seconds

Initial starting points (random):

Final cluster centroids:

Cluster#					
Attribute	Full Data	0	1	2	3
	(1728.0)	(443.0)	(402.0)	(381.0)	(502.0)
buying	1.5	1.8149	1.5572	1.3255	1.3088
maint	1.5	1.8014	1.5572	1.3255	1.3207
doors	1.5	1.6163	1.5149	1.4882	1.3944
persons	1	1.5621	0.3035	0.2231	1.6514
lug_boot	1	1.3679	1.3483	0.727	0.6036
safety	1	1.614	0.2488	1.727	0.508
ClassValues	0.4149	1.3995	0	0.1286	0.0956

Clustered	Instances
0	443 (26%)
1	402 (23%)
2	381 (22%)
3	502 (29%)

As shown above, the four clusters have about the same amount of instances. This indicates that separating the data into the amount of class labels is an efficient method of choosing *k* for the *k*-means run with the car data. What's interesting is that in the raw data, the frequency of the unacceptable rating is significantly higher than, even the combination of the rest of the class labels. However, these results suggest that all of the features may be helpful in determining the car evaluation score.

kMeans - CMC
Number of iterations: 16
Within cluster sum of squared errors: 1058.2965997208046
Time taken to build model (full training data) : 0.02 seconds

Final cluster centroids:

		Cluster#		
Attribute	Full Data	0	1	2
	(1473.0)	(717.0)	(406.0)	(350.0)
wife_age	32.5384	32.1827	32.9754	32.76
wife_ed	2.9586	3.4296	1.9532	3.16
husband_ed	3.4297	3.7824	2.734	3.5143
num_child	3.2614	3.2455	3.6995	2.7857
wife_religion	0.8506	0.8061	0.9729	0.8
employment	0.7495	1	0.9532	0
husband_occ	2.1378	1.9191	2.5764	2.0771
sol_index	3.1337	3.4421	2.399	3.3543
media_expo	0.074	0.007	0.2143	0.0486
ClassValues	0.9199	1.2371	0.399	0.8743

Clustered	Instances
0	717 (49%)
1	406 (28%)
2	350 (24%)

The CMC data shows a different pattern in which the first cluster takes almost half of the data points. The other two clusters are about even in terms of instance distribution. This is quite peculiar because with the *k* choice method that was justified previously, it's assumed that the clusters may represent the class labels in an accurate way. In fact, the first label of CMC, no-use, is the most frequent in the data, and thus the 0 cluster may correspond to that label. However, the fact that the one cluster accounts for almost 50% of the instances, the data may have features that are redundant/not helpful in predicting the class label.

EM

The EM algorithm was allowed to figure out *k* with cross-validation, as opposed to presetting the number of clusters beforehand. This method was chosen because of how EM is trying to improve in a probabilistic metric with a soft clustering (i.e., the data points are more adaptable to be part of one cluster or another). Although it can be justified to use either or method, allowing the algorithm to choose how many clusters are present seems to be a more dynamic approach than presetting a *k*.

EM - CAR

Number of clusters selected by cross validation: 3
Number of iterations performed: 2
Time taken to build model (full training data) : 6.7 seconds
Log likelihood: -5.90744

Attribute	Cluster 0 (0.34)	1 (0.39)	2 (0.27)
buying			
mean	1.7116	1.4176	1.3511
std. dev.	1.088	1.1212	1.1113
maint			
mean	1.6804	1.4253	1.3793
std. dev.	1.0799	1.1257	1.1259
doors			
mean	1.5561	1.4895	1.4444
std. dev.	1.108	1.1179	1.1275
persons			
mean	1.5267	0.1508	1.5469
std. dev.	0.5013	0.3591	0.5291
lug_boot			
mean	1.1091	0.9792	0.8924
std. dev.	0.8025	0.8162	0.8178
safety			
mean	1.6054	1.0007	0.2371
std. dev.	0.4894	0.8067	0.4287
ClassValues			
mean	1.2135	0	0.001
std. dev.	0.797	0.0026	0.0323

Clustered	Instances
0	567 (33%)
1	705 (41%)
2	456 (26%)

Three clusters were found using EM on the CAR data. While the second cluster has the majority of the data points, the instances appear relatively, evenly distributed.

EM - CMC

Number of clusters selected by crossvalidation: 4
Number of iterations performed: 2
Time taken to build model (full training data) : 9.74 seconds
Log likelihood: -4.84805

Attribute	Cluster 0 (0.31)	1 (0.36)	2 (0.13)	3 (0.2)
wife_age				
mean	33.7505	30.8327	35.2739	31.9254
std. dev.	9.5923	7.4571	6.9878	7.1339
wife_ed				
mean	2.4329	3.1192	3.6762	3.0156
std. dev.	1.0173	0.9163	0.5497	1.0117
husband_ed				
mean	3.1178	3.5454	3.8943	3.4012
std. dev.	0.9491	0.694	0.3184	0.8282
num_child				
mean	3.3255	3.5506	2.753	2.9737
std. dev.	2.8214	2.1726	1.6759	2.1747
wife_religion				
mean	0.9702	1	0	0.955
std. dev.	0.1701	0.0029	0.0057	0.2072
employment				
mean	0.8751	1	0.681	0.1458
std. dev.	0.3306	0.0009	0.4661	0.3529
husband_occ				
mean	2.2806	2.1128	1.8948	2.1205
std. dev.	0.8178	0.8841	0.7893	0.9046
sol_index				
mean	2.7702	3.1948	3.7301	3.1976
std. dev.	1.0718	0.9094	0.5266	0.9302
media_expo				
mean	0.1574	0	0	0.1264
std. dev.	0.3642	0.2619	0.0011	0.3323
ClassValues				
mean	0	1.6165	1.0038	1.0414
std. dev.	0.0006	0.5223	0.7844	0.8619

Clustered	Instances
0	449 (30%)
1	660 (45%)
2	206 (14%)
3	158 (11%)

Four clusters were found to be prevalent using EM on the CMC data. The first two clusters appear to have a significant portion of the data, accounting for 75% of the whole set.

Dimensionality Reduction

PrincipleComponents (PCA), IndependentComponents (ICA), and RandomProjection (i.e., RCA) were used for the feature transformation. These were used for the following clustering and ANN algorithms.

k-Means Applied to Dimensionality Reduction

One thing to remember with *k*-means is that the algorithm was not designed to handle nominal data. Unfortunately, the only numerical feature in any of the datasets is wife's age in the CMC set. However, since the data was collected for women who are married, there is not much variance even with this numeric attribute.

_k_-Means – CAR(PCA)
Number of iterations: 31
Within cluster sum of squared errors:
863.8144466153715
Time taken to build model (full training data) :
0.04 seconds

Clustered	Instances
0	468 (27%)
1	411 (24%)
2	378 (22%)
3	471 (27%)

_k_-Means – CMC(PCA)
Number of iterations: 5
Within cluster sum of squared errors:
507.82506729787457
Time taken to build model (full training data) :
0.01 seconds

Clustered	Instances
0	1025 (70%)
1	106 (7%)
2	342 (23%)

k-means for the Car PCA data, again, appears to be relatively evenly distributed amongst the clusters. In the same sense, the first and second cluster for the CMC PCA data remained in order with the first gaining about 20% of the data points. However, the second cluster lost about 21% of its data to the other clusters, which suggest with more iterations, two clusters may be useful in predicting the CMC PCA data.

_k_-Means – CAR(ICA)
Number of iterations: 20
Within cluster sum of squared errors:
327.0758951701447
Time taken to build model (full training data) :
0.03 seconds

Clustered	Instances
0	453 (26%)
1	345 (20%)
2	527 (30%)
3	403 (23%)

_k_-Means – CMC(ICA)
Number of iterations: 9
Within cluster sum of squared errors:
333.2019350665452
Time taken to build model (full training data) :
0.02 seconds

Clustered	Instances
0	511 (35%)
1	629 (43%)
2	333 (23%)

The *k*-means found that the clusters for the CAR ICA are also relatively evenly distributed. So far, this suggests that there is, in fact, four clusters for the CAR data. An

interesting behavior happens with the CMC ICA data in that the clusters are no longer skewed in the first cluster's favor. In fact, the second cluster retains the majority of the data points even though the PCA set suggest that the second cluster is low.

_k_-Means – CAR(RCA)
Number of iterations: 7
Within cluster sum of squared errors: 422.558803727185
Time taken to build model (full training data) : 0.02 seconds

Clustered	Instances
0	411 (24%)
1	288 (17%)
2	555 (32%)
3	474 (27%)

_k_-Means – CMC(RCA)
Number of iterations: 15
Within cluster sum of squared errors: 558.6063320024323
Time taken to build model (full training data) : 0.02 seconds

Clustered	Instances
0	503 (34%)
1	480 (33%)
2	490 (33%)

_k_-means for the CAR RCA data is still relatively the same. However, the CMC RCA data appears to be even more evenly distributed compared to PCA and ICA.

Expectation Maximization Applied to Dimensionality Reduction

EM – CAR(PCA)
Number of clusters selected by cross validation: 3
Number of iterations performed: 2
Time taken to build model (full training data) : 4.94 seconds
Log likelihood: -5.80911

Clustered	Instances
0	567 (33%)
1	228 (13%)
2	933 (54%)

EM – CMC(PCA)
Number of clusters selected by cross validation: 3
Number of iterations performed: 7
Time taken to build model (full training data) : 3.83 seconds
Log likelihood: -11.01353

Clustered	Instances
0	370 (25%)
1	490 (33%)
2	613 (42%)

Three clusters were found with EM on the CAR PCA data with only two iterations. The third cluster was the most prominent. The clusters for the EM on the CMC PCA data seem relatively evenly distributed.

EM – CAR(ICA)
Number of clusters selected by cross validation: 2
Number of iterations performed: 3
Time taken to build model (full training data) : 2.77 seconds
Log likelihood: 14.67844

Clustered	Instances
0	1210 (70%)
1	518 (30%)

EM – CMC(ICA)
Number of clusters selected by cross validation: 2
Number of iterations performed: 4
Time taken to build model (full training data) : 2.4 seconds
Log likelihood: 21.52983

Clustered	Instances
0	629 (43%)
1	844 (57%)

EM on the CAR ICA data finds only two clusters with an interesting distribution. The first cluster has 70% of the data points and the second, the other 30%. This is one of the expected outcomes that makes the most sense because of how the class labels are structured for the CAR data: unacceptable, acceptable, good, and very good. It's not completely clear where the clusters are aligning. However, in the best case scenario, the first cluster represents "unacceptable" and the second represents the rest. This makes sense because an unacceptable car is semantically different than an acceptable, good, and very good car (i.e., you would not want to purchase an unacceptable car, but factors might lead to just choosing an acceptable car). This result can also be alluded to when looking at the distribution of unacceptable cars present in the data (i.e., 70% of the instances are unacceptable).

EM on the CMC ICA data also has two clusters, but it is less clear what those clusters may represent. Although no-use of contraceptive methods is the majority of the instances, it only accounts for about 42% of the raw data (which is pretty close to what the first cluster distribution is).

EM – CAR(RCA)
Number of clusters selected by cross validation: 5
Number of iterations performed: 1
Time taken to build model (full training data) : 6.55 seconds
Log likelihood: 10.41474

Clustered	Instances
0	365 (21%)
1	576 (33%)
3	432 (25%)
4	355 (21%)

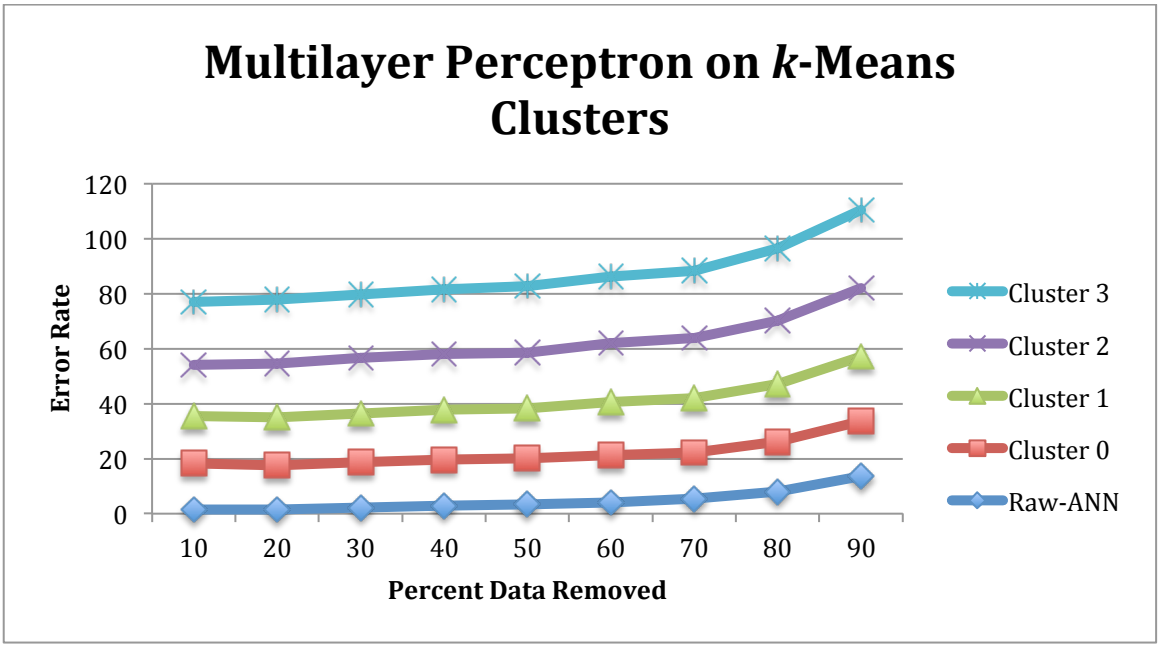
EM – CMC(RCA)
Number of clusters selected by cross validation: 4
Number of iterations performed: 2
Time taken to build model (full training data) : 6.49 seconds
Log likelihood: -13.46311

Clustered	Instances
0	865 (59%)
1	215 (15%)
2	149 (10%)
3	244 (17%)

The clusters for the EM for the CAR RCA returned to being evenly distributed with four clusters. However, it found that the safety feature as no longer being significant. The EM for the CMC RCA also returned to four clusters and the first cluster, again became the most important.

Neural Network Fed with Clustered Data

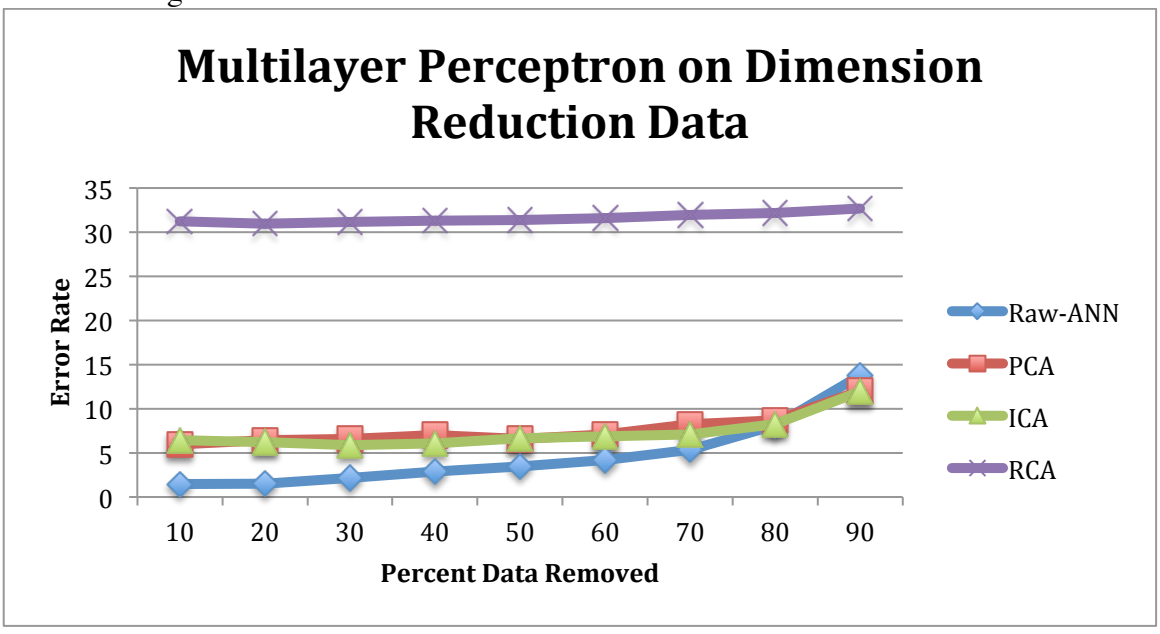
The CAR data was chosen to run the ANN for the clustered and dimension reduction data because ANN performed the best for assignment 1 using the raw data.



As seen above, the first cluster performs better than the next. The best performing cluster still did not do better than when ANN is applied to the raw data.

Neural Network Fed with Dimension Reduced Data

PCA, ICA, and RCA, as well as the original ANN data (represented as Raw-ANN) are presented below. Percent Data Removed represents how much data is allocated to the training set from the total set.



As shown, RCA performed the worst with an average error rate of 31.6%. Interestingly, the three feature selection algorithms performed worse than running an ANN on the raw data. However, there are a few things to consider. The PCA, ICA, and

RCA data were not explicitly split into training and testing sets. The Car Data Reg-ANN actually represents training data that consisted of 80% of the whole set. This may account for the error difference of about 3% between the original ANN and the one ran against PCA and ICA. Considering the type of datasets CAR and CMC are and their relatively low number of dimensions, it's not surprising the ICA and PCA performed better than RCA. RCA is more appropriate for problems with higher dimensions because it addresses issues regarding curse of dimensionality (which isn't as present in these data sets as some others). Still, RCA performs relatively well with an average error rate of 31.6%.

Conclusion

Choosing *k* is an issue in itself when implementing clustering algorithms. There are many different ways in choosing *k* and, unfortunately, I did not have enough time to test different methods. It may have been more useful to allow the algorithms to find the number of clusters itself with *k*-means as opposed to presetting them. This may explain the relatively stable cluster distribution across the feature transformation algorithms.

There is much importance in reiterating that *k*-means was not created to handle nominal values. Although the datasets were converted to numerical values, dummy coding the data does not address the issue. The *k*-means data was somewhat tame for the CAR data, but there were some interesting behavior happening with the CMC data. PCA seems to exaggerate the distribution of data points in the clusters. However, in ICA and RCA's case, they become relatively evenly distributed. This may also explain why the CAR data remained relatively evenly distributed. However, that does not explain why the CMC data had different behaviors.

There are a number of configurations that were left as default for WEKA to handle. For example, the learning rate (for the ANN), iterations, number of attributes, kurtosis calculations. These options were not preconfigured so it should be further explored and may lead to better performance than what was presented above.

The curse of dimensionality is a real concern with unsupervised learning. However, both datasets were relatively noise-free and had a low number of features, which allowed for the feature transformation algorithms to perform well with a low number of instances.

Lastly, the training times for the ANN PCA, ICA, and RCA was significantly less than when training with the raw data (in assignment 1). There are two reasons this may be the case. First, the datasets for the feature transformation algorithms were not split into training and testing sets. This would lead one to believe that it might take significantly more time to train the algorithm. However, since normalization of the data is significant for clustering and feature transformations, it may have drastically shortened the training time. This is because ANN typically takes longer with large numbers, even if there is a low amount of instances present.