Assignment 3 Analysis **gth836x; 901929700**
Myles Lefkovitz **mlefkovitz@gatech.edu**

About the Data:

Dataset 1 - Wine Quality data:

As anyone will tell you, wine's quality is subject to one's taste. Despite that, the University of California Irvine's Machine Learning Repository hosts a Wine Quality Data Set[i] with around 5,000 wines, rated on a 1-10 scale. Each wine has 11 (presumably measured) real-valued attributes.

The 11 attributes and their minimum and maximum values are: fixed acidity [3.8 – 14.2], volatile acidity [0.08 – 1.1], citric acid [0 – 1.66], residual sugar [0.6 – 65.8], chlorides [0.009 – 0.346], free sulfur dioxide [2 – 289], total sulfer dioxide [9 – 440], denisty [0.98 – 1.04], pH [2.72 – 3.82], sulphates [0.22 – 1.08], and alcohol [8 – 14.2]. These numbers may mean something to you if you're interested in wine.

The attributes are all real-valued, therefore they can all be normalized/scaled. We don't need to review them here.

As noted by the data set contributor: the classes are not balanced. There are much more 'normal' wines than great or poor ones. Since quality was rated by humans, there no wines received ratings of 1, 2, or 10. You can see a breakdown of the quantities for each quality rating in *Figure 1* & *Figure 2* to the right.

Quality	Count
3	20
4	163
5	1,457
6	2,198
7	880
8	175
9	5
Total	4,898

Figure 1. Wine Quality Chart

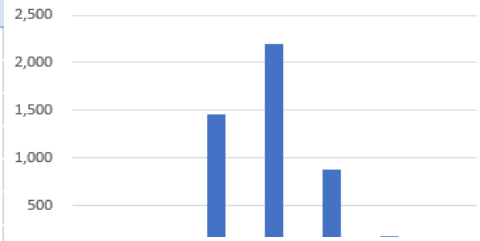


Figure 2. Wine Quality Histogram

If the data set's 7 categories were balanced, we would expect chance to correctly classify a wine 1/7th or around 14% of the time. By reviewing the data, we can see that about 45% of the samples are categorized in the 6/10 quality category. To create a high-performing simplistic model, we could uniformly classify every wine as a 6/10. We should consider a learner successful only if it categorizes wines correctly more than 45% of the time.

Dataset 2 – Adult Income data:

For my second data set I searched for a classification problem that differed from the wine problem above, which had real-valued factors and many classes. On the University of California Irvine' Machine Learning Repository I found an Adult Income Data Set[ii] that classifies adults into one of two income categories: '>50K', or '<=50K'. The '>50K' category identifies individuals that earned more than $50,000 in the given year, 1994. The '<=50K' category identifies individuals that earned less than or equal to $50,000. $50,000 in 1994 is approximately $81,000 in today's terms. The data has 13 attributes, 5 of which are real-valued, and 8 of which are categorical.

The 5 real-valued attributes and their minimum and maximum values are: age [17 – 90], education-num [1 – 16], capital-gain [0 – 99,999], capital-loss [0 – 4,356], hours-per-week [1 – 99].

The 8 categorical attributes and a brief description are: workclass (type of employer), education (level of education, duplicated in the education-num real-valued attribute), marital-status, occupation (type of job), relationship (family status), race, sex, native-country. See figures 3-10 below for an examination of these 8 attributes.

The data set's 2 categories, '>50K' and '<=50K', represent approximately 24% and 76% of the instances in the data set, respectively. To create a high-performing simplistic model, we could uniformly classify every individual in the '<=50K' category. We should consider a learner successful only if it categorizes adult incomes correctly more than 76% of the time.

Summary

These two datasets should prove interesting in the clustering and dimensionality reduction space when compared. The wine dataset has only 11 features, which are all continuous, while the income dataset has 46 features, 41 of which are binary. I'm interested to see how binary features affect the performance of these algorithms.

I. Clustering Algorithms

For both data sets I followed a similar procedure:

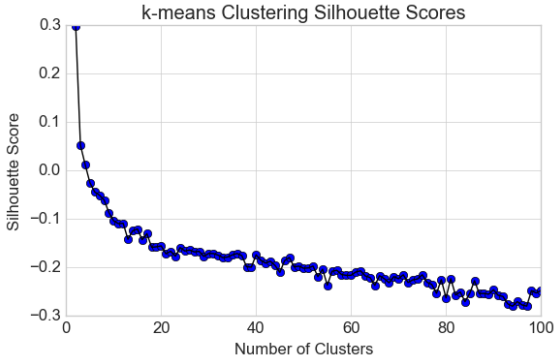
1. Scale and standardize
2. Run through 1-100 clusters and review the results
3. Produce a number of graphs with 2 and 3 clusters

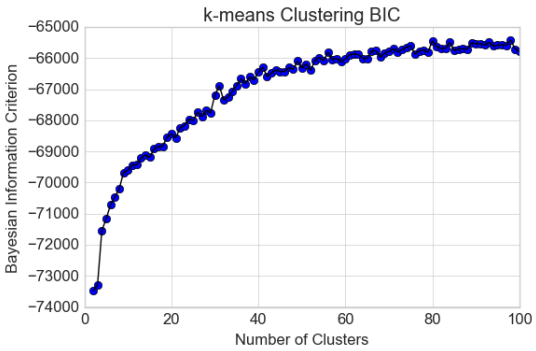
Dataset 1 – Wine

1. k-means

Identifying the optimal number of clusters:

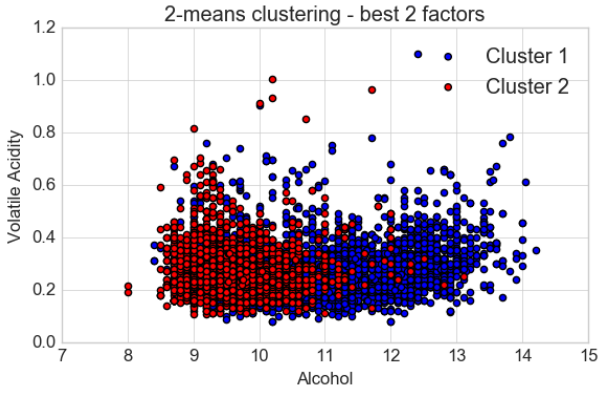
I ran an analysis on silhouette scores and Bayesian information criteria (BIC) to determine the optimal number of clusters with each algorithm. The optimal number clusters should appear where silhouette scores are maximized, or BIC is minimized.

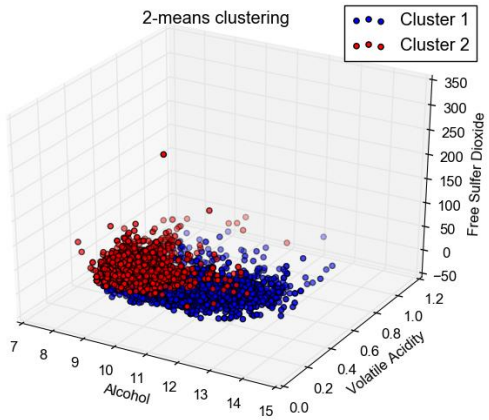




For k-means the optimal clusters is 2.

My next step was to examine these two clusters for insight into the data. I produced many graphs, but the following two are the most clear:



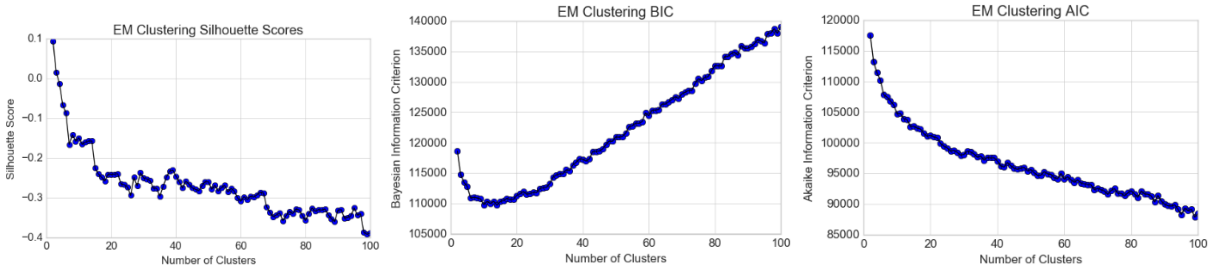


We can see in the plots above that the clusters are highly aligned with the most important feature in the decision tree algorithm, 'Alcohol'. In the charts above, we can see that almost all items in the first cluster have alcohol >11 and almost all items in the second cluster have alcohol <11. The k-means algorithm doesn't appear to be taking volatile acidity, or free sulfur dioxide (the 2nd and 3rd most indicative features identified by the decision tree algorithm) into account in the clustering exercise.

2. Expectation Maximization

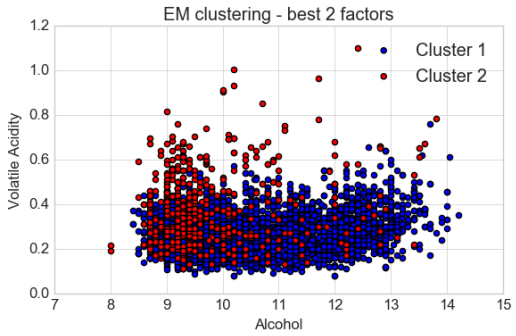
Identifying the optimal number of clusters:

I ran an analysis on silhouette scores and Bayesian information criteria (BIC) to determine the optimal number of clusters with each algorithm. The optimal number clusters should appear where silhouette scores are maximized, or BIC is minimized.



For expectation maximization, the optimal clusters may be 2, 14, or 99

My next step was to examine these two clusters for insight into the data. I produced many graphs, but the following two are the most clear:



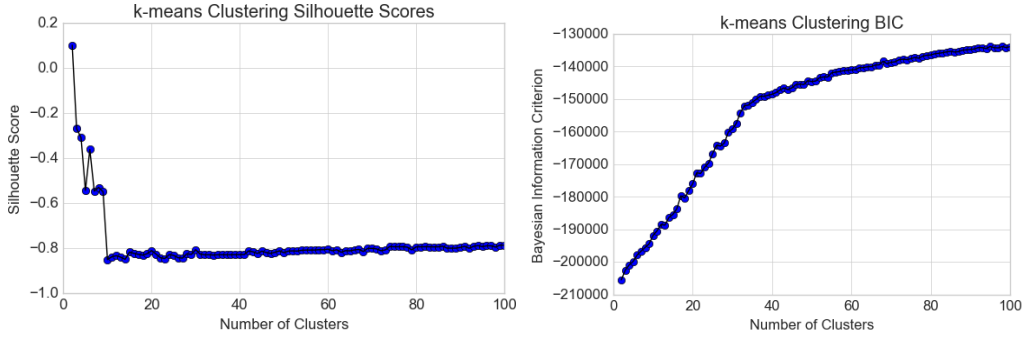
We can see in the plot above that the clusters are highly aligned with the most important feature in the decision tree algorithm, 'Alcohol', but that it seems less restricted than the k-means plot did. In the chart above, we can see that almost all items in the second cluster have alcohol <11, but it's no longer true that all items in the first cluster have alcohol >11. The expectation maximization algorithm appears to be taking volatile acidity, or other features into account in the clustering exercise.

Dataset 2 – Income

1. k-means

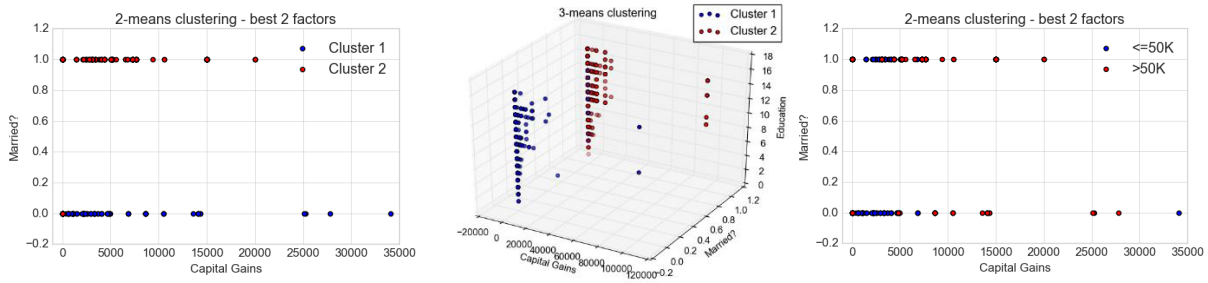
Identifying the optimal number of clusters:

I ran an analysis on silhouette scores and Bayesian information criteria (BIC) to determine the optimal number of clusters with each algorithm. The optimal number clusters should appear where silhouette scores are maximized, or BIC is minimized.



For k-means the optimal clusters is 2.

My next step was to examine these two clusters for insight into the data. I produced many graphs, but the following two are the most clear:

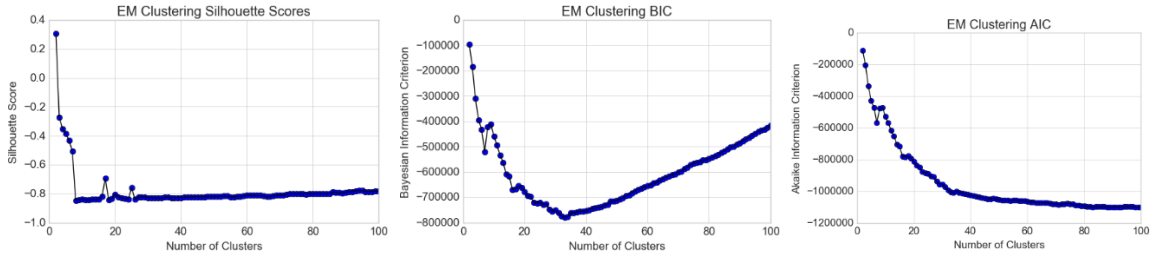


We can see in the plots above that the clusters are highly aligned with the 2nd most important feature in the decision tree algorithm, 'married status' (Y/N). In the first and 2nd charts, we can see that almost all items in the first cluster have marriage status 0 and almost all items in the first cluster have marriage status 1. In the third chart, we can see that there's a strong relationship between marriage, higher capital gains, and high earnings, with red (high earning) dots more prevalent where 'Married?' = 1 and on the right side of the x-axis. The k-means algorithm doesn't appear to be taking capital gains into account in the clustering exercise.

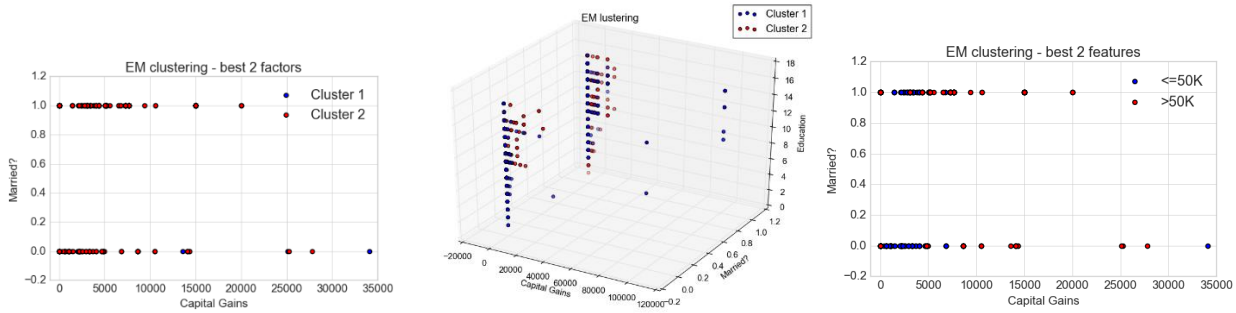
2. Expectation Maximization (EM)

Identifying the optimal number of clusters:

I ran an analysis on silhouette scores, Bayesian information criteria (BIC), and Akaike information criteria (AIC) to determine the optimal number of clusters with each algorithm. The optimal number clusters should appear where silhouette scores are maximized, BIC is minimized, or AIC is minimized.



For expectation maximization, the optimal clusters may be 2, 33, or 98.



We can see that the clusters are better aligned than k-means with the 2 most important features in the decision tree algorithm, 'married status' (Y/N) and 'capital gains'. In the first and second charts, we can see that marriage status does not have as strong of an influence on cluster as it did in k-means above. In the third chart, we can see that there's a strong relationship between marriage, higher capital gains, and high earnings, with red (high earning) dots more prevalent where 'Married?' = 1 and on the right side of the x-axis.

Summary

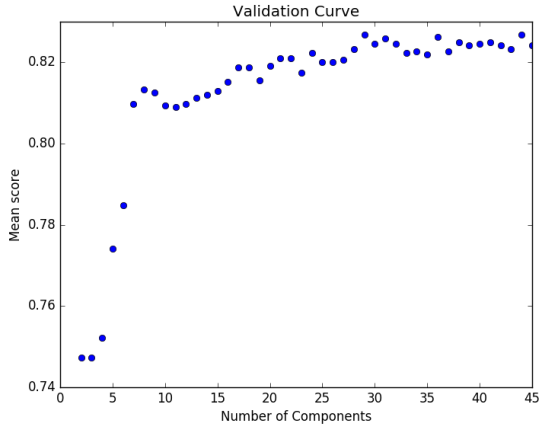
We can see the impact of the probabilistic nature of expectation maximization. It appears to be more willing to create clusters from multiple features.

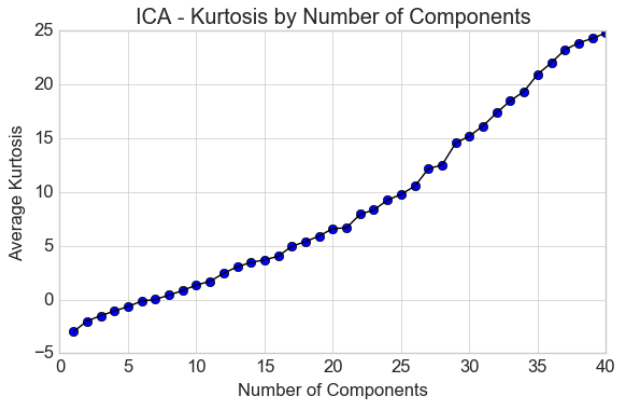
II. Dimensionality Reduction Algorithms

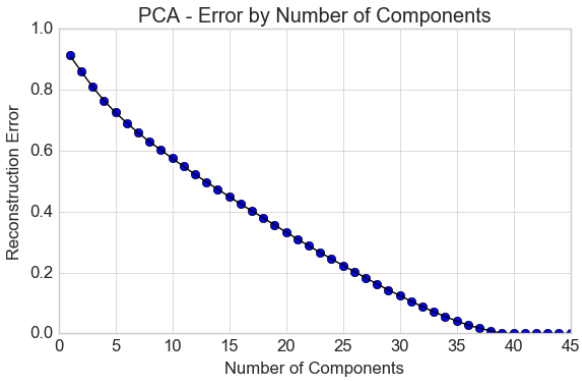
For both data sets I followed a similar procedure:

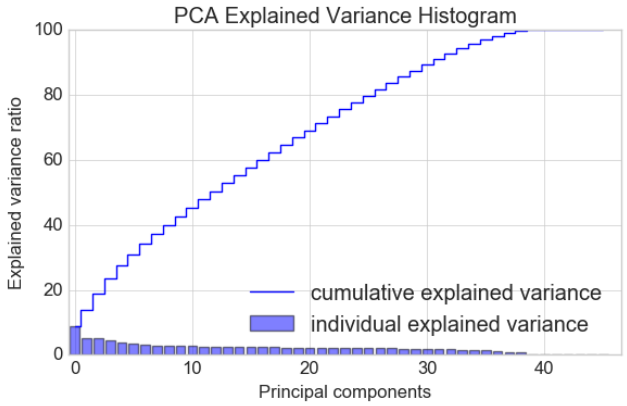
1. Scale and standardize
2. Run through 2-a large number of components (the max number of components was 1 minus the number of features in that dataset – 11 for wine and 46 for income).
3. Use the output of each dimension reduction algorithm with a given number of components as an input for a SVM (support vector machine) learner (similar to the one used in assignment 1) and optimize based on the results.
4. I then checked reconstruction error for PCA, RCA, and SVD and kurtosis for ICA.

In the plot below, you can see an example validation curve, kurtosis plot, reconstruction error curve, and explained variance histogram. These types of charts were used to analyze the efficacy of the dimensionality reduction algorithms.









Dataset 1 – Wine

Dimensionality Reduction	n_components	Mean Score	Training Set Accuracy	Test Set Accuracy	Runtime in s
PCA	10	45.0%	53.4%	51.0%	30
ICA	10	45.0%	51.3%	49.6%	11
RCA	9	54.7%	54.5%	52.3%	16
Truncated SVD	9	52.7%	52.7%	51.0%	29

1. Principle Component Analysis (PCA)

By categorizing 45% of the CV results and 51% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 10 components.

2. Independent Component Analysis (ICA)

By categorizing 45% of the CV results and 49% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 10 components.

3. Randomized Projection (RP)

By categorizing 55% of the CV results and 52% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 9 components.

4. Truncated Singular Value Decomposition (Truncated SVD)

By categorizing 53% of the CV results and 51% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 9 components.

Dataset 2 – Income

Dimensionality Reduction	n_components	Mean Score	Training Set Accuracy	Test Set Accuracy	Runtime in s
PCA	34	84.3%	84.8%	83.8%	72
ICA	29	82.7%	82.9%	83.5%	30
RCA	22	83.6%	84.6%	82.4%	25
Truncated SVD	34	84.4%	84.8%	83.8%	97

1. Principle Component Analysis (PCA)

By categorizing 84% of the CV results and 84% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 34 components.

2. Independent Component Analysis (ICA)

About the algorithms

By categorizing 83% of the CV results and 83% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 29 components.

3. Randomized Projection (RP)

By categorizing 84% of the CV results and 83% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 22 components.

4. Truncated Singular Value Decomposition (Truncated SVD)

By categorizing 84% of the CV results and 84% of the test set correctly with SVM, this dimensionality reduction algorithm performed best with 34 components.

Interestingly, this algorithm had almost exactly the same performance as PCA.

III. Running Clustering Algorithms on Dimensionality Reduced Data

I used the same methodology discussed in part 1 to determine the appropriate number of clusters for each clustering algorithm on each dataset. I used the optimal number of components discussed in part 2 above as inputs and reduced the dimension of the data before running the clustering algorithms.

Dataset 1 – Wine

K-means on reduced dimensions	n_components	# of clusters (Score)	# of clusters (Silhouette Score)	# of clusters (BIC)	Runtime in s
PCA	10	99	2	2	346
ICA	10	100	2	3	450
RCA	9	100	2	2	549
Truncated SVD	9	100	2	3	579

1. k-means + PCA

For k-means + PCA, silhouette and BIC scores agree. 2 clusters is ideal.

Continuing the analysis, the clustering algorithm had a low, 1.2% predictive accuracy rate vs the labels.

2. k-means + ICA

For k-means + ICA, silhouette and BIC scores disagree. I continued the analysis with both 2 and 3 clusters.

Continuing the analysis, the clustering algorithm had a low, 1.2% predictive accuracy rate vs the labels on both 2 and 3 clusters.

3. k-means + RP

For k-means + RPA, silhouette and BIC scores agree. 2 clusters is ideal.

Continuing the analysis, the clustering algorithm had a low, 1.2% predictive accuracy rate vs the labels.

4. k-means + Truncated SVD

For k-means + Truncated SVD, silhouette and BIC scores disagree. I continued the analysis with both 2 and 3 clusters.

Continuing the analysis, the clustering algorithm had a low, 1.2% predictive accuracy rate vs the labels on both 2 and 3 clusters.

EM on reduced dimensions	n_components	# of clusters (Score)	# of clusters (Silhouette Score)	# of clusters (BIC)	# of clusters (AIC)	Runtime in s
PCA	10	100	2	17	100	757
ICA	10	100	3	15	99	687
RCA	9	100	2	13	99	692
Truncated SVD	9	100	2	12	97	569

5. EM + PCA

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 17, and 100 clusters.

Continuing the analysis, the clustering algorithm had a low, 1% predictive accuracy rate vs the labels on 2, 17, and 100 clusters.

6. EM + ICA

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 3, 15, and 99 clusters.

Continuing the analysis, the clustering algorithm had a low, 1%-2.2% predictive accuracy rate vs the labels on 2, 3, 15, and 99 clusters.

7. EM + RP

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 13, and 99 clusters.

Continuing the analysis, the clustering algorithm had a low, 2.4% predictive accuracy rate vs the labels on 2, 13, and 99 clusters.

8. EM + Truncated SVD

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 12, and 97 clusters.

Continuing the analysis, the clustering algorithm had a low, 1.5% predictive accuracy rate vs the labels on 2, 12, and 97 clusters.

Dataset 2 – Income

K-means on reduced dimensions	n_components	# of clusters (Score)	# of clusters (Silhouette Score)	# of clusters (BIC)	Runtime in s
PCA	34	100	2	2	144
ICA	29	100	15	3	152
RCA	22	100	2	2	146
Truncated SVD	34	100	2	2	195

1. k-means + PCA

For k-means + PCA, silhouette and BIC scores agree. 2 clusters is ideal.

Continuing the analysis, the clustering algorithm had a low, 72 % predictive accuracy rate vs the labels. (<76% accuracy is worse than chance)

2. k-means + ICA

For k-means + ICA, silhouette and BIC scores disagree. I continued the analysis with 2, 3, and 15 clusters.

Continuing the analysis, the clustering algorithm had a low, 72% predictive accuracy rate vs the labels on 2, 3 and 15 clusters. (<76% accuracy is worse than chance)

3. k-means + RP

For k-means + RPA, silhouette and BIC scores agree. 2 clusters is ideal.

Continuing the analysis, the clustering algorithm had a low, 59% predictive accuracy rate vs the labels.

4. k-means + Truncated SVD

For k-means + Truncated SVD, silhouette and BIC scores agree. 2 clusters is ideal.

Continuing the analysis, the clustering algorithm had a low, 72% predictive accuracy rate vs the labels.

EM on reduced dimensions	n_components	# of clusters (Score)	# of clusters (Silhouette Score)	# of clusters (BIC)	# of clusters (AIC)	Runtime in s
PCA	34	100	2	48	100	338
ICA	29	100	2	27	91	179
RCA	22	99	2	74	99	206
Truncated SVD	34	100	2	48	100	620

5. EM + PCA

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 48, and 100 clusters.

Continuing the analysis, the clustering algorithm had a low, 72% predictive accuracy rate vs the labels on 2, 17, and 100 clusters.

6. EM + ICA

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 27, and 91 clusters.

Continuing the analysis, the clustering algorithm had a low, 63% predictive accuracy rate vs the labels on 2, 3, 15, and 99 clusters.

7. EM + RP

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 74, and 99 clusters.

Continuing the analysis, the clustering algorithm had a low, 52% predictive accuracy rate vs the labels on 2, 13, and 99 clusters.

8. EM + Truncated SVD

Silhouette, BIC, and AIC scores revealed misaligned results. I continued the analysis with 2, 48, and 100 clusters.

Continuing the analysis, the clustering algorithm had a low, 72% predictive accuracy rate vs the labels on 2, 12, and 97 clusters.

Interesting, performance on Truncated SVD is very similar to PCA on the Income dataset. The optimal components are the same.

IV. Neural Network Learning on Dimensionality Reduced Data

Dataset – Income

I opted to perform this analysis on the Income dataset (formerly dataset 2).

In parts 4 and 5 I used the optimal neural network identified by cross validation in assignment 1: alpha = 0.005 with 3 hidden layers.

In the previous sections I describe optimizing the dimensionality reduction algorithms.

See the results of the analysis in the table below:

Dimensionality Reduction	n_components	Mean Score	Training Accuracy	Test Accuracy	Runtime in s
PCA	34	83.7%	86.1%	84.2%	26
ICA	29	84.2%	84.7%	84.1%	22
RCA	22	81.3%	84.2%	83.3%	22
Truncated SVD	34	83.7%	86.1%	84.2%	24

1. Principle Component Analysis (PCA)

Using the optimal number of components, PCA returned stronger results than the ANN learner did alone, in under 30 seconds (~1% of the time the learner alone took).

2. Independent Component Analysis (ICA)

Using the optimal number of components, ICA returned stronger results than the ANN learner did alone, in under 30 seconds (~1% of the time the learner alone took).

3. Randomized Projection (RP)

Using the optimal number of components, randomized projections returned strong results in under 30 seconds (~1% of the time the learner alone took). These results were not as strong as the learner alone performed in assignment 1.

4. Truncated Singular Value Decomposition (Truncated SVD)

Using the optimal number of components, Truncated SVD returned stronger results than the ANN learner did alone, in under 30 seconds (~1% of the time the learner alone took).

Summary

The curse of dimensionality, however, is very real. It was shocking to see that dimensionality reduction algorithms used before supervised learning could improve the learner's effectiveness over their performance in assignment 1 while increasing their speed dramatically as well.

V. Neural Network Learning on Dimensionality Reduced and Clustered Data

Dimension Reduction	Clustering	n_components	Clusters	Mean Score	Training Accuracy	Test Accuracy	Runtime in s
PCA	kmeans	34	2	74.7%	74.7%	76.3%	2
ICA	kmeans	29	2	74.7%	74.7%	76.3%	2
ICA	kmeans	29	3	74.7%	74.7%	76.3%	12
ICA	kmeans	29	15	78.1%	78.5%	76.9%	21
RCA	kmeans	22	2	78.3%	78.7%	78.4%	7
Truncated SVD	kmeans	34	2	74.7%	74.7%	76.3%	2
PCA	EM	34	2	78.3%	78.3%	79.38	15
PCA	EM	34	48	83.5%	86.0%	84.9%	25
PCA	EM	34	100	82.9%	85.4%	83.7%	31
ICA	EM	29	2	74.7%	74.7%	76.3%	2
ICA	EM	29	27	82.2%	82.5%	81.5%	22
ICA	EM	29	91	83.9%	84.1%	83.8%	29
RCA	EM	22	2	75.1%	75.0%	76.0%	12
RCA	EM	22	74	81.3%	82.1%	80.2%	29
RCA	EM	22	99	82.6%	83.9%	84.6%	31
Truncated SVD	EM	34	2	78.3%	78.4%	79.4%	13
Truncated SVD	EM	34	48	83.5%	86.0%	84.7%	23
Truncated SVD	EM	34	100	83.0%	85.5%	84.2%	30

1. k-means + Dimensionality Reduction

In the previous sections I identified the highest scoring number of components for each dimensionality reduction algorithm and the ideal number of k-mean clusters for each input. As you can see in the grid above, there's little variation in the performance of the neural network learner with k-means. ICA + k-means with 15 clusters and randomized projection + k-means with 2 clusters perform the best, with scores above 78%. As mentioned in the introduction, because of the nature of the data, algorithms with scores below 76% perform worse than chance.

Runtimes were especially low for some of these algorithms, but the decreased cost isn't enough to make up for the lost performance.

2. Expectation Maximization + Dimensionality Reduction

Running the neural network learner on data that had been dimensionally reduced by one of the algorithms and again by expectation maximization yielded far better results. Some results (PCA with 34 components + EM with 48 clusters) outperformed the neural network learner on raw data, with test accuracy of 84.9% (the highest ANNs had achieved in assignment 1 was 83.5% test set accuracy; SVMs were able to go as high as 83.9%). On top of this increased accuracy was a dramatic decrease in runtime. The ANN in assignment 1 ran for 49 minutes. The ANNs ran against our reduced datasets in under 30 seconds each time.

Summary

Given labeled data, the clustering algorithms weren't able to perform as well as the supervised learning algorithms we experimented with in the first mini-course. The curse of dimensionality, however, is very real. It was shocking to see that clustering and dimensionality reduction algorithms used in tandem before supervised learning could improve the learner's effectiveness over their performance in assignment 1 while increasing their speed dramatically as well.

---

[i] http://archive.ics.uci.edu/ml/datasets/Wine+Quality

[ii] http://archive.ics.uci.edu/ml/datasets/Adult