CS 4641
Hardik Sangwan

Assignment 3: Unsupervised Learning

(Dataset and Choice Description from Assignment #1)

Dataset #1: Tic-Tac-Toe End Game

This database encodes the complete set of possible board configurations at the end of tic-tac-toe games, where "x" is assumed to have played first. The target concept is "win for x" (i.e., true when "x" has one of 8 possible ways to create a "three-in-a-row").

Data Set Instances – 958;
Number of Attributes – 9; Missing Attribute Values – None;
[Additional Info in Dataset files]

Dataset #2: King-Rook vs. King-Pawn on a7 End Game

The pawn on a7 means it is one square away from queening. It is the King + Rook's side (white) to move. The dataset contains 37 attributes. The first 36 describe the board and the 37^{th} attribute has a 'win' 'nowin' value.

Data Set Instances – 3196;
Number of Attributes – 37; Missing Attribute Values – None;
[Additional info in Dataset files]

Dataset Choice: What makes this interesting?

The key things I was looking for in my datasets were –

1. The ability to easily understand and contextualize the data –
   Both of the above datasets are games I understand, with simple String Data describing their boards and a 'Win/NoWin' Binary classifier.

2. The ability to perform multiple experiments with different parameters –
   This meant that I didn't want to work with datasets that were too huge to run without taking up too much time. Both these datasets have around a 1000 to 3000 instances and none of the algorithms took more than a few minutes to run.

3. Getting results that were worthy of comparisons –
   I had to go by trial and error to look at a few different datasets and finally decide on the ones that fared reasonably differently on error and performances metrics. The ones above did so I went with them.

Clustering

Tic-Tac-Toe Data Set

k-means randomly separates the data into k clusters. Distance between data and centers is iterated as new means for clusters are calculated after reassigning points to clusters. This reassignment continues until a maxima is reached for the required k clusters. The algorithm was run at different values of k to find minimum number of incorrectly clustered instances. As number of clusters are increased beyond number of labels, incorrectly clustered instances increase. Interestingly, the lowest value is found at just 1 cluster because 65 % of the data is 'positive' leading to a 35 % error in clustering. (Obviously this would generalize very poorly)

k	Error (%)
2	49.478
3	55.428
6	69.103
7	72.238

Once the value for k has been determined as 2, seed value was changed to account for getting stuck at a local maxima. From the iterations we can see that the error value oscillates with the best value found at 300.

Seed	Error
10	49.478
100	46.346
200	46.24
300	28.91
500	48.37
1000	37.36

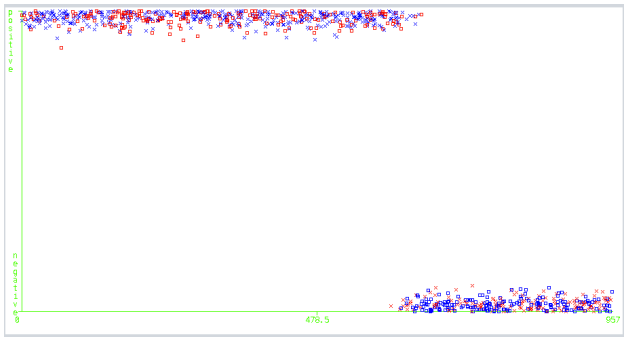
Notes –
All tests run at time close to 0 sec.
No noticeable performance change between Euclidean and Manhattan distance.
No change for increasing number of max iterations, meaning that the algorithm for k=2 and seed = 300 requires very few iterations (even maintain the same value of 28.91 at just a single iteration)

Expectation Maximization (EM) calculates the likelihood of a point belonging to a cluster and then updates the means with weighted probability values and reiterates. The process continues till convergence. Here is error with number of clusters.

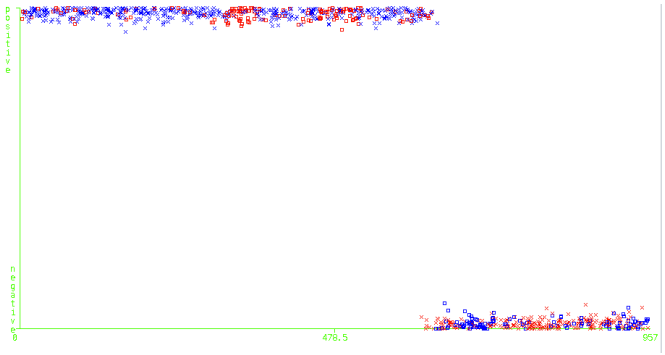
Clusters	Error (%)	Time to Build (s)
Cross Validation (4)	63.25	10.25
2	42.172	.18
3	60.75	.24
5	69.83	.35

Decreasing threshold value (from .01) leads to decreased error up to a certain point (1E-4) of convergence after which it starts to flat out at 42.172 %. Changing the number of max iterations and seeds also does not affect the error value, meaning that the algorithm is not getting stuck at any local maxima and this is the peak value it can achieve for this dataset.

Clusters at optimal values



k-Means



EM

Both similar diagrams (because of the two optimal clusters) show the clusters with k-Means showing a better classification of points among the clusters.

Kr-vs-kp Data Set Optimal Values (Found similarly to the Tic-Tac-Toe data set)

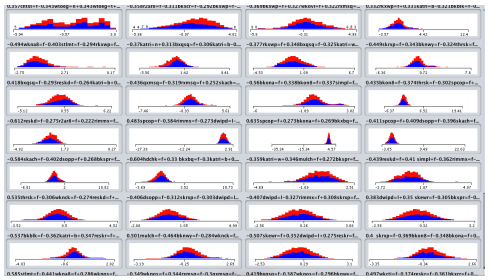
k-Means – 31.66 % error @ k = 2, seed = 700, no effect from iterations, distance type

EM – 39.1427 % error @ k = 2, no effect from iterations, threshold, seed.

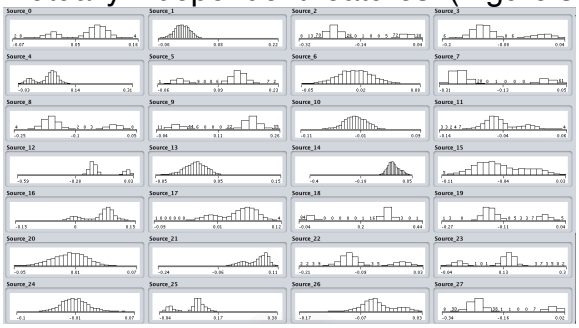
Clustering with Dimensionality Reduction

Dimensionality reduction can help in modeling data by reducing the number of attributes. This problem can be approached in different ways, as showcased by the four algorithms used in this assignment.

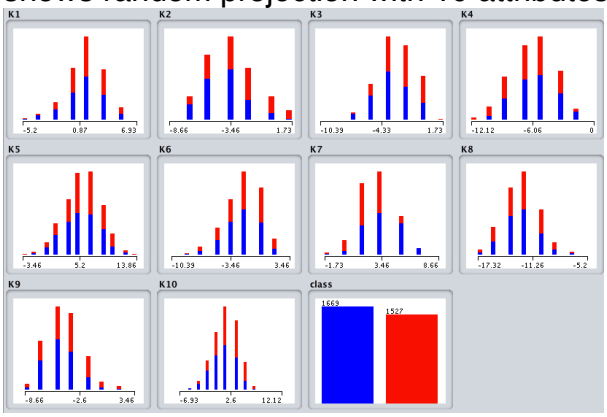
Principal Component Analysis maps data in a way to maximize variance to ensure the most amount of information is captured. (Figure below shows data after applying PCA to kr-vs-kp)



Independent Component Analysis assumes that the data is the output of many unobserved sources. It makes a linear transformation from current feature space to new space to create mutually independent features. (Figure shows ICA applied to kr-vs-kp)



Random Projection takes the current data and projects it onto random directions (Figure shows random projection with 10 attributes)



Information Gain ranks attributes based on how much unique information they contribute to the dataset.

Kr-vs-kp Endgame Results

PCA (k = 2)

Algorithm	Variance	Incorrectly Clustered Instances %	No. of Attributes
k-Means	.25	48.686	4
EM	.25	48.529	4
k-Means	.45	48.74	9
EM	.45	45.685	9
k-Means	.65	48.52	15
EM	.65	42.55	15
k-Means	.85	48.24	25
EM	.85	44.93	25
k-Means	.95	48.84	32
EM	.95	39.73	32

Large threshold for variance allows capturing most information, leading to lower error values for EM. K-Means on the other hand shows little change for different variance values meaning that it does not gain any relevant information from increase in attributes. Eigenvalues for the attributes range from .8 to .04.

ICA (attributes = 37)

Algorithm	k	Incorrectly Clustered Instances %
k-Means	2	48.87
EM	2	48.87
k-Means	3	43.08
EM	3	52.409

k-Means results in lower error for 3 clusters after ICA. This could be caused by the features now becoming independent and as ICA focuses more than on the parts than the whole as compared to PCA, particular attributes that are more important lead to lower errors if 3 clusters are used.

Randomized Projection (k = 2)

Algorithm	Seed	Number of Attributes	Incorrectly Clustered Instances %
k-Means	42	11	49.405
EM	42	11	46.15
k-Means	42	21	49.53
EM	42	21	45.33
k-Means	100	21	41.52
EM	100	21	41.17
k-Means	100	26	40.76
EM	100	26	41.11
k-Means	100	26	40.70
EM	100	26	40.17
k-Means	1000	26	47.309
EM	1000	26	47.34
k-Means	500	26	44.18
EM	500	26	44.27
k-Means	200	26	49.84
EM	200	26	48.9

While overall values tend to fluctuate for both the number of attributes and seed value selected, the bet values found were for seed = 100 and number of attributes = 25.

IG (k = 2)

Attributes were removed based on their ranking through the WEKA attribute evaluator

Algorithm	Incorrectly Clustered Instances %	No. of Attributes
k-Means	41.33	10
EM	38.54	10
k-Means	42.95	20
EM	39.14	20
k-Means	35.7	25
EM	39.14	25
k-Means	46.37	30
EM	39.14	30
k-Means	31.66	35
EM	39.14	35
k-Means	31.66	37
EM	39.14	37

Shows that k-means needed all the attributes to be effective and lead to its lowest error but EM maintains its value throughout different attribute numbers, meaning it is getting its important probability information from just a few attributes

Tic-Tac-Toe Results

PCA (k = 2)

Algorithm	Variance	Incorrectly Clustered Instances %	No. of Attributes
k-Means	.95	36.74	17
EM	.95	36.84	17
k-Means	.85	30.06	15
EM	.85	36.84	15
k-Means	.65	36.74	11
EM	.65	36.74	11
k-Means	.45	36.74	7
EM	.45	36.74	7
k-Means	.25	49.37	5
EM	.25	49.58	5

Lower variance values lead to lower overall information about the feature space. As opposed to k-Means, EM maintains its error at lower attributes. Similar results about attributes are seen through IG

IG (k = 2)

Attributes were removed based on their ranking through the WEKA attribute evaluator

Algorithm	Incorrectly Clustered Instances %	No. of Attributes
k-Means	28.91	10
EM	42.17	10
k-Means	46.34	8
EM	42.17	8
k-Means	47.70	6
EM	48.32	6
k-Means	48.12	5
EM	47.19	5

Again, EM shows less dependence on the attributes than k-Means which has a significant decrease in accuracy as attributes are reduced.

Randomized Projection (k = 2)

Algorithm	Seed	Number of Attributes	Incorrectly Clustered Instances %
k-Means	42	11	46.76
EM	42	11	47.8
k-Means	100	11	48.85
EM	100	11	48.95
k-Means	42	8	48.53
EM	42	8	46.03
k-Means	100	8	48.43
EM	100	8	48.25
k-Means	42	6	49.37
EM	42	6	44.46
k-Means	100	6	48.43
EM	100	6	46.24

Basically produces the same results for most different measurements (close to random chance)

Dimensionality Reduction Conclusion for Clustering

Overall, the best parameters for dimensionality reduction lead to minor decrease in error (for e.g., from 39.14 to 38.54 for EM through IG for kr-vs-kp). But in the remaining three algorithms, even the best parameters lead to an increase in error. This maybe because there is a better tuning for the parameters that hasn't been selected or they suffer too much from the curse of dimensionality so loss of attributes is leading to loss of information.

Neural Net with Dimensionality Reduction and Clustering

Kr-vs-kp Endgame Data Set Results (Dimensionality Reduction)

Algorithm	Correctly Classified Instances %	Best Parameters	Model Building Time	Testing Time
No Reduction	99.8	-L .1 -M .5 -H a	23.41	.04
PCA	99.9	Variance - .95	14.97	.02
ICA	99.9	Default	19.05	.03
Random Projection	100	Seed - 100 Attributes - 26	10.17	.07
IG	98.03	Attributes - 25	11.47	.03

Neural Net takes less time to run as dimensions are reduced, as is the case for random projection and IG, but no significant improvement or loss is seen when these reduction algorithms are applied using their best parameters to an already highly accurate training model.

Kr-vs-kp Endgame Data Results (with Dimensionality Reduction and Clustering, k = 2)

Reduction Algorithm	Clustering Algorithm	Correctly Classified Instances %	Best Parameters	Model Building Time	Testing Time
No Reduction	-	99.8	-L .1 -M .5 -H a	23.41	.04
PCA	k-Means	99.87	Variance - .95	15.22	.02
ICA	k-Means	99.87	Default	20.49	.03
Random Projection	k-Means	99.96	Seed - 100 Attributes - 26	10.46	.02
IG	k-Means	99.27	Attributes - 25	12.88	.03
PCA	EM	99.90	Variance - .95	15.11	.02
ICA	EM	99.87	Default	20.23	.03
Random Projection	EM	99.84	Seed - 100 Attributes - 26	10.51	.02
IG	EM	98.12	Attributes - 25	12.1	.03

Just as seen previously, high accuracy is maintained for a lower model building time when these algorithms are applied.