**CS7641 Unsupervised Learning HW3**

Aleksandr Plokhikh ([aplokhikh3@gatech.edu](mailto:aplokhikh3@gatech.edu))

**Introduction, dataset selection, data preprocessing**

The goal of the assignment is study unsupervised learning algorithms such as clustering, and feature transformation + selection, and study how the last can help with supervised learning (NN), and how substitution of the labels with the ones from clustering affects on supervised learning and possible fixes preexisted mislabeling.

For the task I take the same fruits [1], and phones dataset [2] which are interested for a number of reasons which described in A1, but here for A3 we have a new dimension of interest. Fruits have pretty determined classes of different fruits with genes of the fruits, but in case of phones, the cell phones just divided on 4 price ranges, and we do not know how arbitrary is the division on classes, we can easily do more classes, and do not break any logic, but it will be nice to se what is the reasonable number of price ranges(clusters there).

I did data preprocessing for both datasets (linearly scaled all features from 0 to 1 where 0 -is the minimal feature value in the set, and 1 – is the maximum). This is extremely important step for A3. It is very important for determining distances for KMeans, reasonable expectation calculations for EM, and very important for some feature transformation techniques especially crucial for PCA! For example for phones dataset variance of the phones in battery capacity in mAh is 2-3 orders magnitude higher than difference in CPU clock speed in GHz, and there is obvious difference in importance for customer. For example average consumer may not see difference between phones with 3502mAh and 3500mAh difference but corresponding difference between 3.5Ghz, and 1.5GHZ of CPU clock speed will be visible as day and night especially if we take into account that phone with 3.5Ghz CPU most likely will have more cores, and definitely different architecture with wider pipelines which are not usually listed in phones spec list but have a very substantial multiplier on performance even further than simple ratio of 3.5Ghz/1.5GHz.

If we will not implement scaling then PCA on phones dataset will decide that principal components are battery capacity in mAh, mass of phones in grams, and length of it in mm or some linear combination of them since there are obvious positive covariance of this parameters. They will dominate the process while they are not the most crucial features. Clustering will be also spoiled and will divide phones on clusters based on battery capacity, weight, and length ignoring other aspect with Euclidean or Manhattan distancing in Kmeans or in EM. The same situation will be in fruits dataset but with other parameters like mass, and length of the fruit. Based on my knowledge I can parse reasonable scaling of the parameters but it ruins the idea unsupervised learning, I can not pass domain knowledge to the algorithm, and just do the best most simple knowledge independent scaling of each parameter from 0 to 1.

**Part 1, Clustering**

On fruits dataset most simple unsupervised standard square error (SSE) performance measure method show that with increasing of number of clusters for KMeans method of clustering gradually decreases, which does make perfect sense. The plot does not show any noticeable inflection point. SSE is not good method for the database. Unsupervised Silhouette method on contrary shows very prominent inflection on number of clusters equal to 11 for KMeans, and 10 for EM. Any further increase in clusters number does not make sense especially for EM algorithm since the metric does not go lower anymore. The supervised homogeneity score does show inflection at number of clusters equal to 7 (the real number of clusters) for both methods, the homogeneity score is also high overall which says that both clustering method produces clusters cvery close to the right labes. The larger number of clusters offered by Sihoutte method may say that for larger fruit types (linear size and wheigh) very is a some variation itraclass variation in size which causes etra clustering. Small variation of big values in one class for KMeans, and EM are as significant as many variation of smal parameter in different classes. It wmast be the issue of datascaling but I did not have a right to offer better scaling method without parsing additional domain of knowledge to the algorithm. KMean algorithm ran for maximum 200 iteration and 10 restarts while EM ran for maximum 100 iteration and 5 restart and even with much smaller number of possible iterations EM took 531ms, while KMeans only 65ms. EM is much more expensive algorithm to run and high dimentiality of features data make things even worse.

A group of graphs with numbers and lines

Description automatically generated with medium confidence

Fig. 1 KMeans and EM performance of fruits and phones datasets.

On the phones dataset SSE graph has a prominent knee at n =5, and less prominent at n=9, so the most reasonable clusterrring according to SEE for phones is n-5 since it is simpler clustering we use Okkams razor here and do not mumtiply instansies withoutneed, anso inflection on n-5 is more distinct than at n=9 which look more like small diviation from the trend. Silhouette method proposes clustering with 4, and 3 classes for KMeans and EM respectively. Any increase of number of clusters leads to increase of the Silhouette score -not efficient. The mean decision here ( for KMeans SSE, KMeans Sile, and EM Silhouette) is 4 which correspond to the number of classes in dataset so the number of clasees ready does make sence here. The Homogenity score is almost monotonically rises with rise of clusters, but remains very low (less than 0.02) it may say that both KMeans, and EM divides dataset on some clusters which are not assotiated with price. The low Homogenity level may be caused by noise in the data, and the fact that the actual pricing of the phones are not always objective, but depends on the marketing, brand position, current market situation, and we do not have such info in the database. KMeans algorithm takes 32ms to run on the dataset while EM takes 92ms for 4 clusters spilt. There is 3-fold difference in time but it is not as dramatic as in case of fruits dataset where we had 9 times difference for 10 clusters. Most likely this difference directly arises from higher dimentiality of drits data (34 vs 20) and higher number of clusters to split in case of fruits data, since EM is the “soft clustering” technic and for each point we need to calculate probability relation to each cluster and if we have roughly 3 times more clusters it means that we need 3 times more time!

**Part 2 Dimension reduction techniques**

PCA, and ICA are well established algorithms, and they perform equally well on both on both sets of data in terms of reconstruction error, a minor difference between decompositions occurs when the number of algorithm features becomes equal to the number of features and reconstruction error drops to lowest values but this setting does not make to much sense since we do not reduce number of features, noise and do not fight the dimensionality problem here which says that we need to exponentially increase number of entities in training sample to get information out of it.

The reasonable number of features of fruits dataset for PCA and ICA I chose to be 20 since at this setting the reconstruction error already drops bellow 10^-4 and it offeres very reasonable features reduction over default 34 dimentions in the original dataset.. At n around 30 reconstruction error settles at values of 10^-13 which is exceptionally low and tell about very high efficiency of methods on the dataset In case of the pones dataset the reconstruction error drops not that rapidly but more in linear way and after set of 17 features the recostraction error does not fall dramatically so I chose this number to still have benefit over 20 dimentions in the original dataset. Reconstruction error of both methods on phones dataset is orders magnitude higher than on fruits dataset which tesll us that bot PCA and ICA are not as exeptionaly good here as on fruits dataset. It may happened because phones database have a lot of noise, not clear, and non-linnerar relations within the data. PCA takes 16ms of run time on both datasets, ICA -13ms on fruits dataset, and surprisingly less than 1ms on phonses dataset, it may only say that the ICA decomposition there is very straightforward eventhough the reconstruction error around 10^-3.

Reconstruction error for random projection error graducaly decreas with rise of dimensions number dimention. The reconstruction error is many times higher than in case of PCA and ICA methods for both datsets. For fruits datset the difference is order magnitudes. The reconstrcuction error as in the case of PCA, and ICA drops to the lowest values when we reach the demention of the original dataset. For all fweature transformation methods I sed the same number of dimentios (20 for fruits,a nd 17 for phones) for better apple to apples comparison. to have comparable nosie, and convolution recution capabilities in later parts of the assignment. The random projection method unsurprisingly takes less than 1ms to conduct calculations

A group of graphs showing different types of numbers

Description automatically generated

Fig. 2 Feature transformation methods performance

I tried all manifold learning techniques and all of them are very times consuming, and often requires ordersof magnitude longer computation times than PCA or ICA. It happens for LTSA, MDS, Spacial Embeding, and TSNE, that is why they did no come in the short list as impractical. Hessian embading was not only very slow but for comparable with PCA, ICA dimention number equal to 20 for fruits data set and 17 for phones datset we had to calculate embading for a very large number of neighbors it is not only very time consuming but for fruits dataset calculation of the neighbour list already takes a significant portion of all entries and for 20 components the neighbiurs of the entry is already more than half of the all enties.

LLE is the fastest manifold learning but still it takes 420ms on fruits database and 1.15s for phones which is higher than PCA or ICA run times, while performance is poor. Isomap showed similar performance as LLE but ran 3 times longer than LLE. I chose MLLE with 50neighbour calculation since it increases calculation time very insagnificantely in comparison with default LLE (to 466ms and 1266ms for fruits and phones datasets respectively), while increasing performance. Still despite all efforts MLLE did not beat PCA and ICA on my datasets. On Fruits dataset MLLE went neck to neck until number of components rose to 15 after that point PCA,a dn ICA continued to fast decrease of reconstruction error while MLLE could not maintain the same pace. On Phones database MLLE was as good as PCA, and ICA, and for low componnents numbers of 2-6 even offered better reconstruction errors but, such low diminations are hardlipractical since they have a high reconstruction error of more than 0.05, and number of components may not be enough for sophisticated further learning algorithms to retrieve information. It is worth mentioning that when we increase components number to number of features in the original dataset the reconstruction error did not drop to the very low error it may be the cause of non linearity and locality of the MLLE, it may not provide a very low reconstruction error on the whole space of the data due to the fact.

In short PCA, and ICA are equally good on both dataset, ICA takes less time to learn, RP maybe the fastest but high reconstruction error makes its implementation not justifiable PCA, and especially ICA on phones dataset does not take a lot of time to process. MLLE does not provide performance boost comparing to PCA/ICA but takes much more time to process. Other manifold techniques take even more time or show worse performance without significant speed up gain.

**Part 3, Clustering of processed datasets**

In this part of the taks it is intrestin to see how different dimietion reduction techniques helmps with clustering or makes it more difficult. Does it help or prevent finding the most appropriate number of clusters. Alo comparison will be presented on Fruits dataset since on the untreated dataset the Silhoulle score very nicely showed the best clustering approach, and I will treat it as my best benchmark, and it will be also nice to see if feature transformation techniques will offer a right number of clusters equal to 7 (classes in original dataset) instead of EM+Silhoutte result of 11 clusters. All dimension reduction algorithms reduced components from 34 to 20 for better performance comparison.

A graph of a number of data

Description automatically generated with medium confidenceA graph of a number of fruits

Description automatically generated with medium confidenceA graph of a number of data

Description automatically generated with medium confidenceA graph of a number of objects

Description automatically generated with medium confidence

Fig. 3 EM clustering on fruits dataset after dimensionality reduction

The most mediocur result showed MLLE features transformation method. Silhoutte score does not clearly shows optimal clustering, maybe n=9, or n=6 would be ideal dimensions count. Silhouette score even rises with increase of clusters number, which is a sign of poor and inefficient clustering. While homogeneity score is close to 0.75 for high dimensions it does not show the same number of clusters as number of labels in the original dataset equal to 7 and votes more for n=9. Clearly MLLE is not a winner here and do not grasp the essence of the set despite reasonable reconstruction error in part 2. EM algorithm with such data transformation takes longest time to build the “clustering curve presented on Fig.3 (18.7 seconds) which says that EM algorithm needs more iterations and resturts to come to convergence.

Second least performing algorithm here is ICA it also does not have a good “knee” inflection on the Silhouette graph and “votes” more towards to n=9 or n=4 which close to MLLE predictions. Silhouette score here does not rise with the increase of the clusters number which is good sign. The Homogenity curve predicts 6 clusters which is much closer number to the number of reals classes in the data set equal to 7. The homogeneity sscore in not highest overall but respectibale and remains around 0.7 at high clusters number. The EM algorithm takes a bit less time to build the “clustering” curves – 16.7s which says that EM algorithm comes to convergence a bit faster than in case of MLLE pretreated input data.

RP is just a random linear combination of dimensions hence the performance of clustering should be similar to the clustering on the original dataset but slightly worse due to lower number of dimensions and due to building linear relationships between components which most likely are not present in the dataset, and indeed than Silhouette graph have a knee, as an original EM Silhouette graph on the untreated dataset but a bit less prominent and on a slightly higher clustering number equal to 12, instead of 10 as in original clustering. Silhouette score does not rise with te increase of clusters number, which indicates reasonable clustering. The homogeneity curve predicts 6 clusters the same way as in case of ICA , which is still 1 less than the number of true classes in the data set. The homogeneity score keeps minor uptrens at higher clusters numbers and reaches values higher than 0.7. The EM run time is even lower than in case of MLLE, and ICA pretreatments and equal to 14.5 for building a full “clustering curve”. The EM algorithm comes to convergence here faster than in case of ICA, and MLLE algorithms.

PCA is the winner here. Silhoutte groph has a very prominent inflection point at n=11, which is alsost the same as in the EM clustering with original data where was equal to 10, it is also equal to the best KMeans clustering on the untransformed database. Homogeneity score predicts the same number of clusters as true classes, the score rises even further with increase of clusters numbers and reaches the values of close to 0.8 at higher clusters numbers, which are signg of good clustering. The EM algorithm takes the losest time to buld the curve (11.3s) which significately lower than any other aforementioned features transformation techniques, and gap to the closest copetitior – RP is larger than the gap between 2nd and 3rd place and 3rd and 4th plases. It indicates that it is much easier for EM algorithm to come to convergence and produce goo clustering than with any other clustering techniques. The run time of EM algorithm is even lower than on unprocessed dataset which was 13.7s for the full “clustering” curve, but such comparison does not make sense since the original dataset has 34 dimentions while PCA teatead only 20 which reduces calculation time of a single step of any clustering algorithm on the dimention reduced dataset, but it hints that on average EM algorithm need less iterations and restarts to find good converged clustering than with any dimention reduction techniques.

I want to point out the the performance of the EM clustering on Fruits data set increases in the raw MLLE, ICA, RP, PCA, and the clustering run time monotonicaly decreases. The good preprocessed data gives better clusters and it happens faster as well, we sooner reach the convergence. The second thing which I wasn to point out the the same quality outcomes as well as run-time reslts I see alwo on KMeans method on Fruits database, and EM, KMeans algorithms on Phones datasets. In all combinations of clustering techniques and datsets. Performance of chosen algorithm increases in the raw MLLE, ICA, RP, PCA, while the run time decreases, and we can conllude here that combined poduim looks like: MLLE-4th place, ICA- 3rd place, RP- 2nd, PCA-1st for all important metrics (performance, and runtime) regardless of the weight of the metric in the total score.

**Part 4, Neural net performance on density reduced datasets**

One of the main points of features transformation and dimension reduction is to help other learning techniques with dimensity problem which says that we have exponential rise of required data entries with the rise of features count if we want to extract information out dataset.

I performed this task on fruits dataset since in=t has more clasees 7 vs 4 in phones datset hence it should be more difficult task for network and underlining dimension reduction algorithms to produce good result. Also fruits data base have many (34) components and not many entries (898) to properly sittisfy the deminsity of the datset so it is a proper candidate for dimention reduction pretreatment. I used 25x25 network with constant learning rate of 0.01 and “relu” activation function since it was the best parameters for the NN on the untreated database and our dimention reduction algorithms need to fight against this benchmark. All dimension reduction algorithms reduced components from 34 to 20 for better performance comparison.

The learning curve of NN on untreated database clearly shows dimension prombel of the set -lurning process is “rough” with two falls on the validation sample learning curve which indicates that NN learns noise as true data. There is no much room for improvement with the increase of the entries data since the validation curve almost reached the train learn curve. It also indicates that NN learns a lot of noise, and each individual feature may not be ideal. The weighted average f score of the NN on the test sample is 0.8876. Overall NN shows all the signs than it needs dimensions reduction.

PCA algorithm on the dataset proved to provide the lowest reconstruction error on pair with ICA, and was the best helping algorithm. The validation sample learning curve have only one small performance drop with overall rise of the f-score during learning process. It is the sign that NN learn real information and less nose. Overall performance of the validation sample curve is higher than the original untreated one which also a good sign of PCA usefulness here. Tangible difference between validation and training smples curves indicate that NN has a good room for performance growth with extra training data but wven with the current training set the performance of the NN on the test sample is higher that of NN on untransformed input data and weighted average f sore is equal to 0.9214, which prove the usefulness of dimensity reduction for fruits datset.

A graph of a graph of a graph

Description automatically generated with medium confidenceA graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generatedA graph of a graph with numbers and lines

Description automatically generated with medium confidence

A graph of different colored lines

Description automatically generated with medium confidenceA graph of a graph with numbers and lines

Description automatically generated with medium confidence

Fig. 4 Learning curve of NN on density reduced fruits datasets from left to right (untreated set, PCA, ICA; RP, MLLE)

ICA was one of the best in terms of reconstruction error, but did not help EM clustering. The validation sample learning curve very smooth and uprising which indidcates that ICA does a good job eliminating noise dimensions. The f-score on the validation sample is higher in general than the one the original NN which indicates about better performance of the NN with such dimidtions reduction treatment. Unfortunately NN does not have much room for performance increase with additional training data since the the gap between training and validation sample lurning curves closes up at high training sample sizes, but it still manages to beat the original NN on the test sample with the weighted average f-score of 0.9124 which is slightly behind the PCA performance.

RP learning curves very closely resembles the ones for original NN including overall shape, drops and the numbers, and the performance on the test sample is also very close- the f-score ois 0.875 -insignificantely lower than in case of the original NN (lower info for NN most likely influenced this small score reduction). The results it does make sense since RP creats just random linear combination on features, it does not make some oriented sophisticated features transformation hence we should not expected some radical changes in the performance of the new NN, and we did not get it.

MLLE showed reasonably low reconstruction error but failed on the clustering test. The learning curves are smooth like in any “sophisticated” fatures transformation techniqyues such as PCA, and ICA, which telss us that MLEE does help NN not to learn noise in data and focus on reasonable information. The NN trained on MLLE have a limited room for improvement with additional data (based on proximity of train and test sample learning curves). Unfortunately MMLE has only marginal positive effect on NN performance on the test sample where it showed weighted f-score of 0.889 over 0.8876 for original NN, such boost is statisticaly insignificant taking the size of the test sample equal to 180 entries, and dose not justify time spent to dataset transformation (466ms) while the NN took only 93ms to train.

Timewise dimintions reductions does not simplify the NN sagnificantely since it removes only (34-20)\*25=280 connections, while original NN has 34\*25+25\*25\*(25-1)\*25\*7=16025 connections. All NN train in about 91-93ms including the original one, except of PCA “pretreated” NN which takes only 71ms to train which indicates that NN conmes to convergence faster and this fact rorrelates well with the fact the PCA NN showed the best performance and largest area for improvement.

Part 5, NN on dimensions reduced datasets with unsupervised labels

A graph of a graph with blue and orange lines

Description automatically generated