**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 dimention 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 devision on clasees, 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 fro both datasets (linearly scaled all features from 0 to 1 where 0 -is the minima featurel value in the set, and 1 – is the maximum). This is extremely important step for A3. It is very important for determining distancies 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 everage consumer may not see difference between phones with 3502mAh and 3500mAh difference but corresponding difference between 3.5Ghz, and 1.5GHZ of CPU clok 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 linerar combination of them since there are obvioce positive covariance of this parameters. They will dominate the process while they are not the most crucial features. Clustering will mbe also spoiled and will divide phones on clusters based on battery capacity, weight, and length ignoring other aspect with euqlidian or Manhattan distancing in Kmeans or in EM. The same situation will be in fruits datset 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 damain 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 noiticible inflection point. SSE is not good method for the database. Unsuppervised Silhoutte 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 sence 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.