1.5em0pt

CS 6150: "Dimensionality Reduction Project"

Amey Desai, Mina Ghashami, Jared Rose

1 Introduction

Typical databases used in data mining applications may contain millions of records and thousands of variables. In such high dimensional spaces, firstly there would be certain highly correlated variables, inclusion of which in data mining models leads to derivation of inaccurate results, secondly due to their huge volume, data objects appear dissimilar in many ways which makes common data organization strategies (such as clustering) ineffective.

In these situations it is often beneficial to reduce the dimension of the data in order to simplify the data model and improve the efficiency and accuracy of data analysis. They project data from original high dimensional space R^n to a new lower dimensional space $R^k (k < n)$, in which tasks such as classification or clustering often yield more accurate and interpretable results.

Dimension reduction techniques can be broadly classified into feature extraction and feature selection categories. Feature extraction ones, through the application of an either linear or non-linear mapping, produce a new more compact set of dimensions while attempting to preserve the characteristics of data in the original feature set.

Principle Component Analysis (PCA) transforms a data set such that the maximum variance among data objects is preserved. In multimedia data analysis, Latent Semantic Analysis (LSA) which is a variant on PCA, is popular. LSA can reveal semantic information from document co-occurrences, and is based on singular value decomposition (SVD) of the termdocument matrix.

Locality preserving projections (LPP) are linear projective maps that preserve the neighborhood structure of the data set. Theoretical analysis has shown that LPP is an optimal approximation to LDA according to discrimination power. Locality preserving indexing (LPI) uses LPP for document representation and indexing. LPI is an unsupervised approach which discovers the local geometric structure of document space and, obtains a low rank approximation of the space.

In this report, we study three dimension reduction methods, namely PCA, LDA and LPI and three clustering algorithms including Kmeans, Dbscan and Expectation Maximization (EM). We explore the effect of these DR methods on clustering accuracy and efficiency by clustering data before and after dimenion reduction.

Having this fact in mind that according to Kriegel, Kroger and Zimek (2009), clustering in high dimensional space suffers from four problems:

1. Multiple dimensions are hard to think in, impossible to visualize, and, due to the exponential growth of the number of possible values with each dimension, complete enumeration of all subspaces becomes intractable with increasing dimensionality. This problem is known as the curse of dimensionality.

2. The concept of distance becomes less precise as the number of dimensions grows, since the distance between any two points in a given dataset converges. The discrimination of the nearest and farthest point in particular becomes meaningless.

3. A cluster is intended to group objects that are related, based on observations of their attribute's values. However, given a large number of attributes some of the attributes will usually not be meaningful for a given cluster. This is known as the local feature relevance problem: different clusters might be found in different subspaces, so a global filtering of attributes is not sufficient.

4. Given a large number of attributes, it is likely that some attributes are correlated. Hence, clusters might exist in arbitrarily oriented affine subspaces.

2 Related Works

There exist some clustering based dimension reduction methods in literature. Distributed LSI partitions information sources regarding conceptual domains and indexes each sub collection with LSI. Zhang et al. [6] analyze the relation between truncated SVDs of a matrix and the truncated SVDs of its sub-matrices. In [7], authors propose a spherical k-means algorithm for clustering high dimensional and sparse document vectors. They partition the document space to k disjoint clusters and each cluster is represented by a concept vector. The original document matrix can be projected to the concept space spanned by the constructed matrix. Gao et al. [8] propose a clustered SVD strategy for large data sets. They cluster a large inhomogeneous data set into several smaller subsets on which they apply the truncated SVD strategy.

CLSI [9] is a methodology for matrix representation and information retrieval. It first clusters the term-document matrix and then executes partial SVD on each cluster. The extracted information is used to build low-rank approximations to the original matrix.

3 Basics

In this section we give a brief review of three dimension reduction and three clusteing algorithms we intend to implement.

3.1 Dimension reduction algorithms

As dimension reduction algorithms we implement PCA, LDA and LPI.

3.1.1 Locality Preserving Indexing (LPI) [1]

Let $\chi = \{x_1, x_2, \ldots, x_m\}$ be the set of m document vectors, which constitute the document space. Each document x_i, is represented as a u-dimensional vector, $x_i = (x_1^i, x_2^i, \ldots, x_u^i)$. The set of terms constitutes the dimensions, and the value of each dimension for a document x_i, determines the importance of that term in x_i with respect to other terms.

Let X represent the $u \times m$ term-document matrix whose columns are document vectors. Note that in high dimensional spaces the intrinsic dimensionality of document space may be very smaller than u. LPI aims to find a new representation of the document set, $Y = \{y_1, y_2, \ldots, y_m\}$, such that $\|y_i - y_j\|$ reflects$\|x_i - x_j\|$. It first executes a preprocessing step and projects the document set into the PCA subspace by discarding the smallest principal components.

Reason: Our reason for picking LPI is that this method preserves the local structure of a dataset through transformation to a lower dimensional space. In particular, it is able to detect the most representative features in document related data instead of the most discriminant features. It is our belief that this method of data reduction will produce very similar results on both spaces when used in conjunction with clustering.

3.1.2 Principal Component Analysis (PCA)

To find principal components that depict the highest variance directions, PCA involves the computation of mean, $\bar{x}$ and covariance matrix S of the data as shown in the following equations. This is followed by the eigenvector decomposition of S and taking in only the top K eigenvectors corresponding tot he top K eigenvalues.

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{1}$$

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T \tag{2}$$

3.1.3 Linear Discriminant Analysis (LDA) [3]

LDA is a method for identifying the "classification" of individuals based on a series of explanatory variables. This is achieved by uncovering a transformation that maximises between-class separation and minimise within-class separation. To do this, it defines two scatter matrices, S_B for between-class separation and S_W for within-class separation:

Process K-means(χ, k):
  Define k clusters $\mathbf{C} = \{C_1, \ldots, C_k\}$
  Generate k random initial cluster centroids $\mu_1, \ldots, \mu_k$
  repeat
    $C_j = \{x | x \in \chi \,\&\, \forall i \neq j.\, \delta(x, \mu_j) \leq \delta(x, \mu_i)\}$
    $\mu_j = \frac{\sum_{x \in C_j} x}{|C_j|}$
  until $\forall j. |\mu_j^{new} - \mu_j^{old}| \leq \epsilon$

Figure 1: The kmeans algorithm

$$S_B = \sum_{c \in C} n_c (\mu_c - \mu)(\mu_c - \mu)^T \tag{3}$$

$$S_W = \sum_{c \in C} \sum_{j: y_j = c} n_c (x_j - \mu_c)(x_j - \mu_c)^T \tag{4}$$

Where n_c is the number of objects in class c, μ is the mean of all examples and μ_c is the mean of all examples in class c. LDA's objection can be combined into a single maximisation called Fisher Criteria:

$$W_{LDA} = argmax_W \frac{|W^T S_B W|}{|W^T S_W W|} \tag{5}$$

Reason: Our reason for picking LDA is that LDA focuses on data classification over feature classification. We hope that its approach to finding linear combinations of variables that best describe the data will help better define its structure for clustering.

3.2 Clustering Algorithms

As clustering algorithms we implemented Kmeans clustering, DBscan clustering and Expectation Maximization clustering.

3.2.1 K-means

K-means considers documents to be placed in a u-dimensional metric space with an associated distance metric δ. It partitions the data set into k clusters, where k is a user defined parameter. Each cluster C_l, has a centroid μ_l, defined as the average of all data assigned to that cluster. The algorithm relies on finding cluster centroids by trying to minimize the within-cluster sum of squares:

$$\sum_{j=1}^{k} \sum_{x_i \in C_j} \delta(x_i, \mu_j) \tag{6}$$

The formal definition of k-means is given in Figure 3.2.1. The algorithm proceeds heuristically; a set of random centroids are picked initially, to be optimized in later iterations.

Reason: K-means clustering is great at identifying circular and spherical clusters in data. Though simple in design, it is still very effective at clustering a wide range of data.

3.2.2 DBSCAN [4]

DBSCAN is a density based clustering algorithm. Its definition is based on two parameters, ϵ, which is a given distance, and *minpnt*, which is the minimum number of points required to form a cluster. The clusters are constructed based on the concept of density reachability and minPoints. Consider δ as a distance metric.

A point x_i is directly density reachable from a point x_j, if $\delta(x_i, x_j) \leq \epsilon$ and x_i is surrended by atleast *minpnt* points such that one may consider x_i and x_j to be part of a cluster. Also point x_i is called density

reachable from x_l if there exists a chain of data objects $x_i = x'_1, x'_2, \ldots, x'_z = x_l$, such that each two consecutive objects x'_j and x'_{j+1} $(1 \leq j \leq z-1)$ are directly density reachable.

In order to clustering data, it startes with an arbitrary starting point (that has not been visited), retrieves its ϵ neighborhood, and if it contains *minpnt* number of points, a cluster is started. Otherwise, the point is labeled as noise. This point might later be found in a sufficiently sized ϵ environment of a different point and hence be made part of a cluster.

Reason: Unlike K-means, DBSCAN is able to find different-shaped clusters without needing to guess at the number of clusters that will be in the data. This will allow us to cluster data that would otherwise be incorrectly clustered by K-means.

3.2.3 EM Clustering

EM iteratively uses MAP to cluster data points by finding latent variables in the data. EM uses two alternating steps: an expectation step and a maximization step. During the expectation step, the expected value of the log likelihood function is calculated using the results from the previous distribution (the initial values can be chosen arbitrarily). During the Maximization step, the new values found during the expectation step are used to estimate a new distribution that better fits the data. These steps are repeated until the results converge.

Reason: EM Clustering is very resistant to the effects of noise in the data. Furthermore it is able to handle a high dimensionality of data and is not limited to spherical clusters like K-means.

4 Evaluation Metrics

The following metrics are suggested for assessing reciprocal effect of dimensionality reduction on clustering:

4.1 Mutual Information (MI) [5]

Let C denote the set of clusters obtained from original data set and C' obtained from data set after reducing its dimension. Their mutual information $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j). \log_2 p(c_i, c'_j) p(c_i) p(c'_j) \tag{7}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that an object arbitrarily selected from the corpus belongs to clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that an arbitrarily selected object belongs to the clusters c_i as well as c'_j at the same time. Normalized mutual information which is more used in literature is defined as follows:

$$\bar{MI}(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \tag{8}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. $\bar{MI}(C, C')$ ranges from 0 to 1. $\bar{MI} = 1$ if the two sets of clusters are identical, $\bar{MI} = 0$ if the two sets are independent.

4.2 Accuracy (AC) [5]

Assume that data is clustered prior and after dimension reduction. Given a document x_i, let c_i and c'_i be the corresponding cluster label respectively. AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{m} \delta(c_i, map(c'_i))}{m}, \tag{9}$$

Where m is the total number of documents, $\delta(x, y) = 1$ if $x = y$, otherwise $\delta(x, y) = 0$. map(c'_i) is the permutation mapping function that maps each cluster label c'_i to some appropriate cluster label from the original data space. The best mapping can be found by using the Kuhn-Munkres algorithm [**?**].The AC measure basically measures number of documents which are located in the same cluster before and after dimension reduction.

4.3 Rand Index (RI)

The goal of this metric is to assign two documents to the same cluster if and only if they are similar.
A true positive (TP) decision assigns two similar documents to the same cluster, a true negative (TN) decision assigns two dissimilar documents to different clusters. There are two types of errors we can commit. A (FP) decision assigns two dissimilar documents to the same cluster. A (FN) decision assigns two similar documents to different clusters.
The Rand index measures the percentage of decisions that are correct: $RI = \frac{TP+TN}{TP+TN+FP+FN}$

5 Implementation

We implemented most of algorithms in matlab, except for EM clustering which we used a java framework namely ELKI to do that. In implimentation we reduced each dataset to 2 , 6 and 10 dimensional space. In clustering with kmeans, we set the parameter k as number of classes in each dataset. As in clustering with dbscan we used trial and error approach for finding minpoint, while we computed pairwise distance between datappoints of the dataset and used average of that number as an estimation for ϵ (the radius for dbscan). For Mnist dataset, due to its huge size we were unable to find a good radius as running dbscan on it never ended. For Em clustering algorithm, we set the parameter k to the number of classes in each dataset. Specification of the data sets used in evaluation are given in table below.

Table 1: Data sets used in Evaluation

Data set	**Records**	**Dimension**	**Classes**	**Radius used in dbscan**	**Minpoint used in dbscan**
Gisette	6000	5000	2	20800	3
Madelon	2000	500	2	720	3
Olivette	400	4096	40	1500	5
Mnist	60000	784	10		

6 Evaluation Results

6.1 Accuracy

The accuracy diagrams for four datasets when reduced by LPI is shown in figure 2. This diagram is showing the accuracy of data classification with respect to classl labels of the data and clustering result in the target space. Note that the last dimensional space is the original data space.

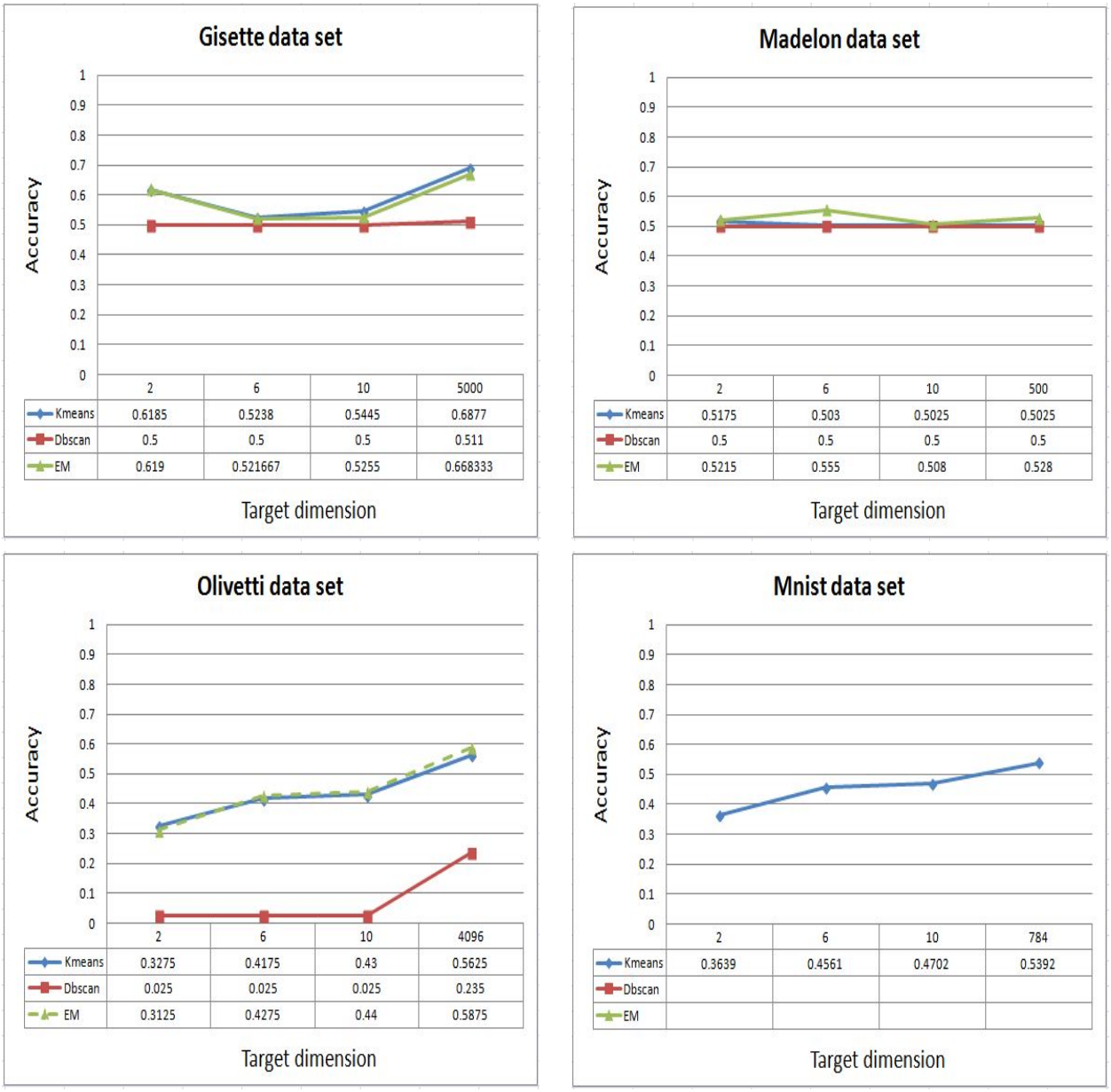

Figure 2: Accuracy of datasets in original and reduced dimension space when reduced by LPI

The accuracy diagrams for four datasets when reduced by LDA is shown in figure 3.

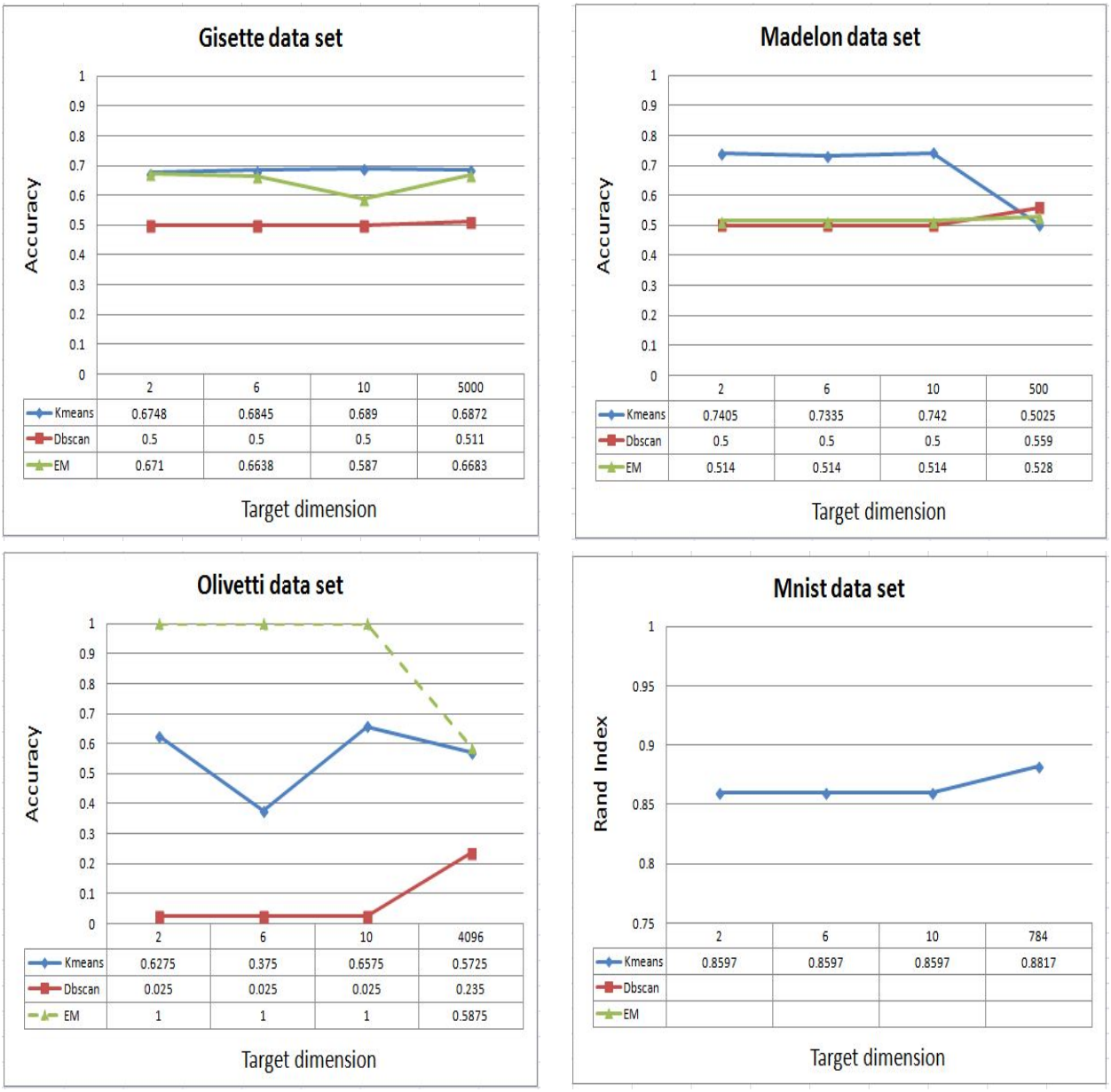

Figure 3: Accuracy of datasets in original and reduced dimension space when reduced by LDA

The accuracy diagrams for four datasets when reduced by PCA is shown in figure 4.

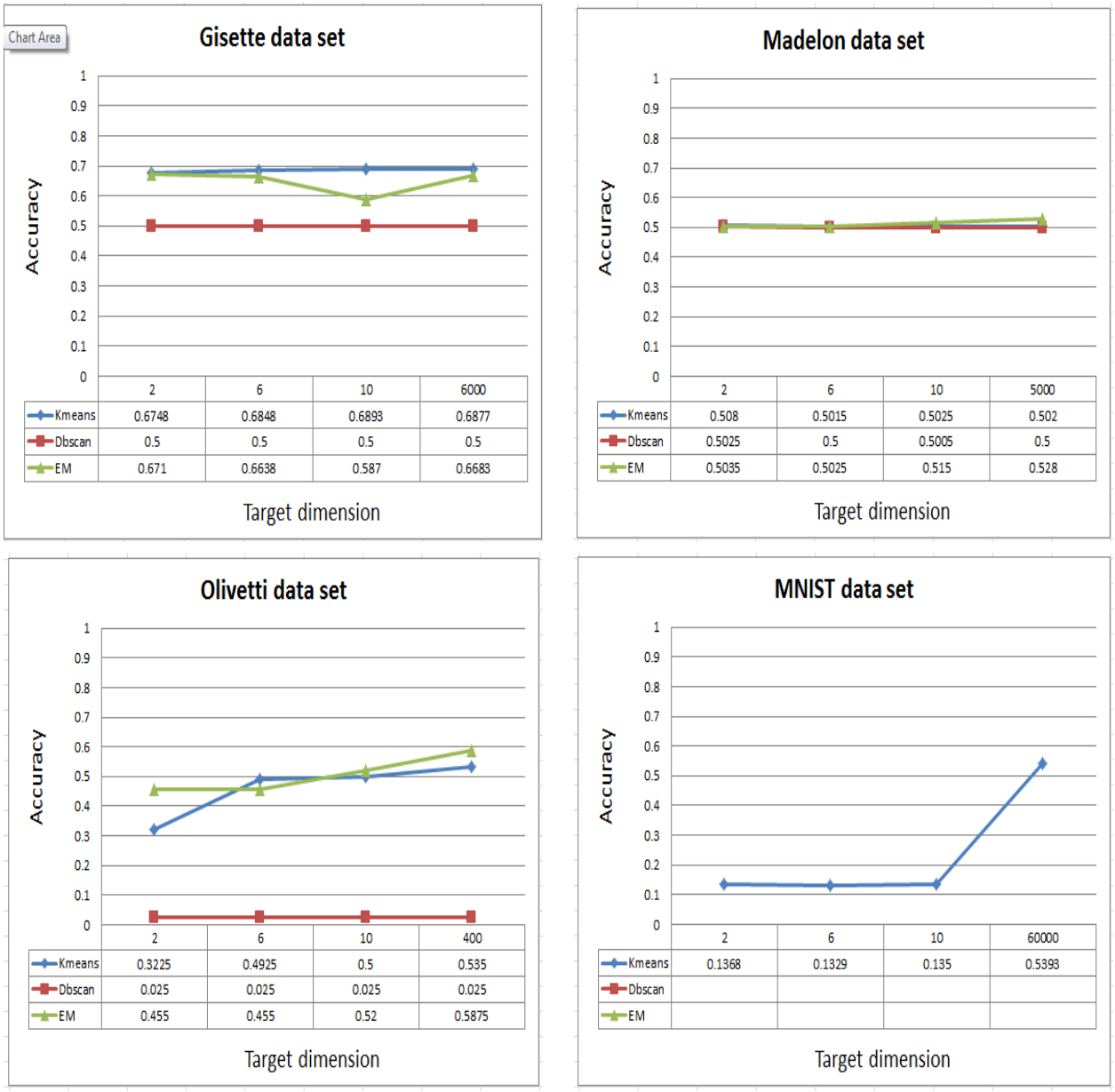

Figure 4: Accuracy of datasets in original and reduced dimension space when reduced by PCA

6.2 Mutual information

The MI diagrams for four datasets when reduced by LPI is shown in figure 5.

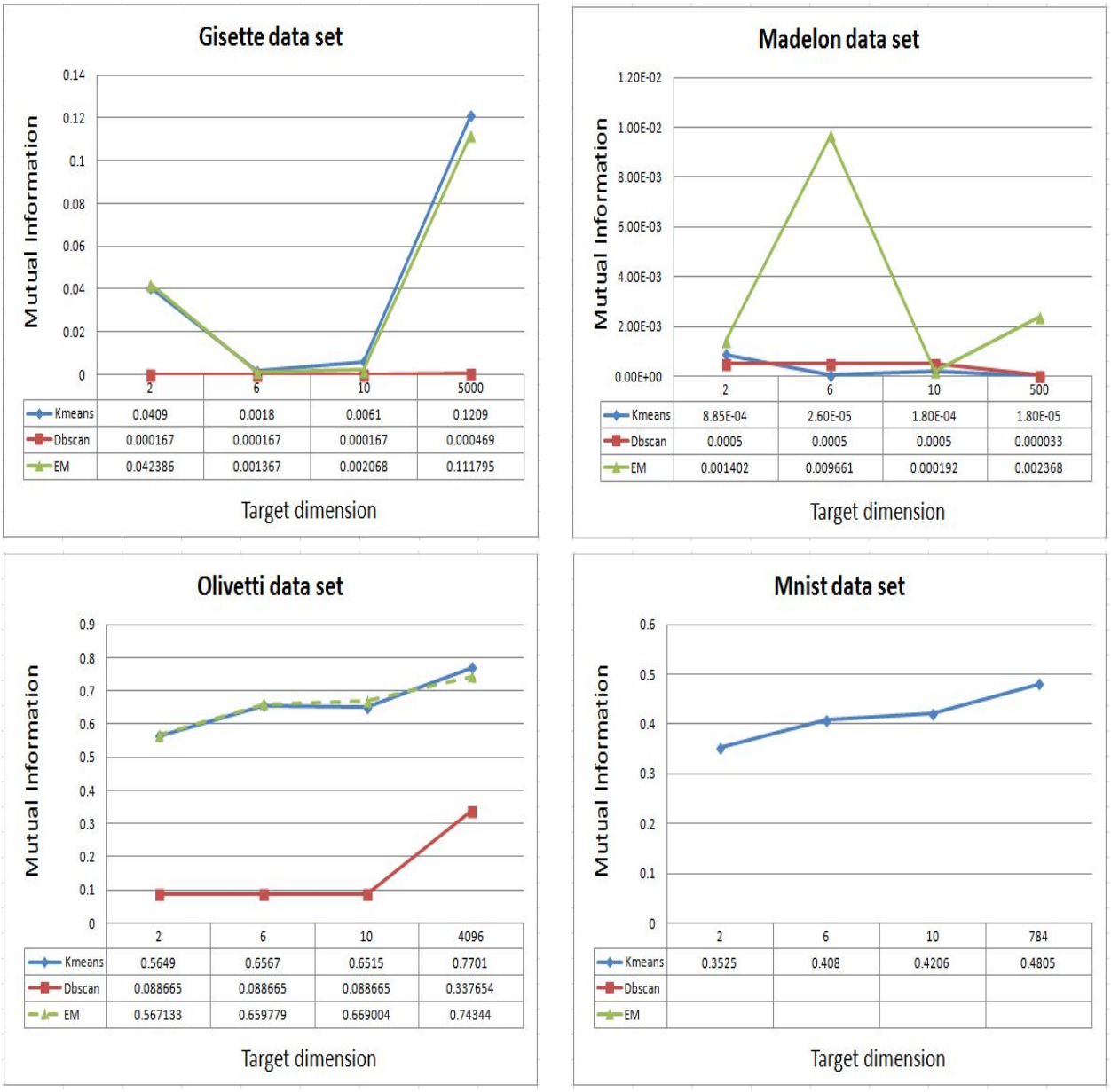

Figure 5: MI of datasets in original and reduced dimension space when reduced by LPI

The MI diagrams for four datasets when reduced by LDA is shown in figure 6.

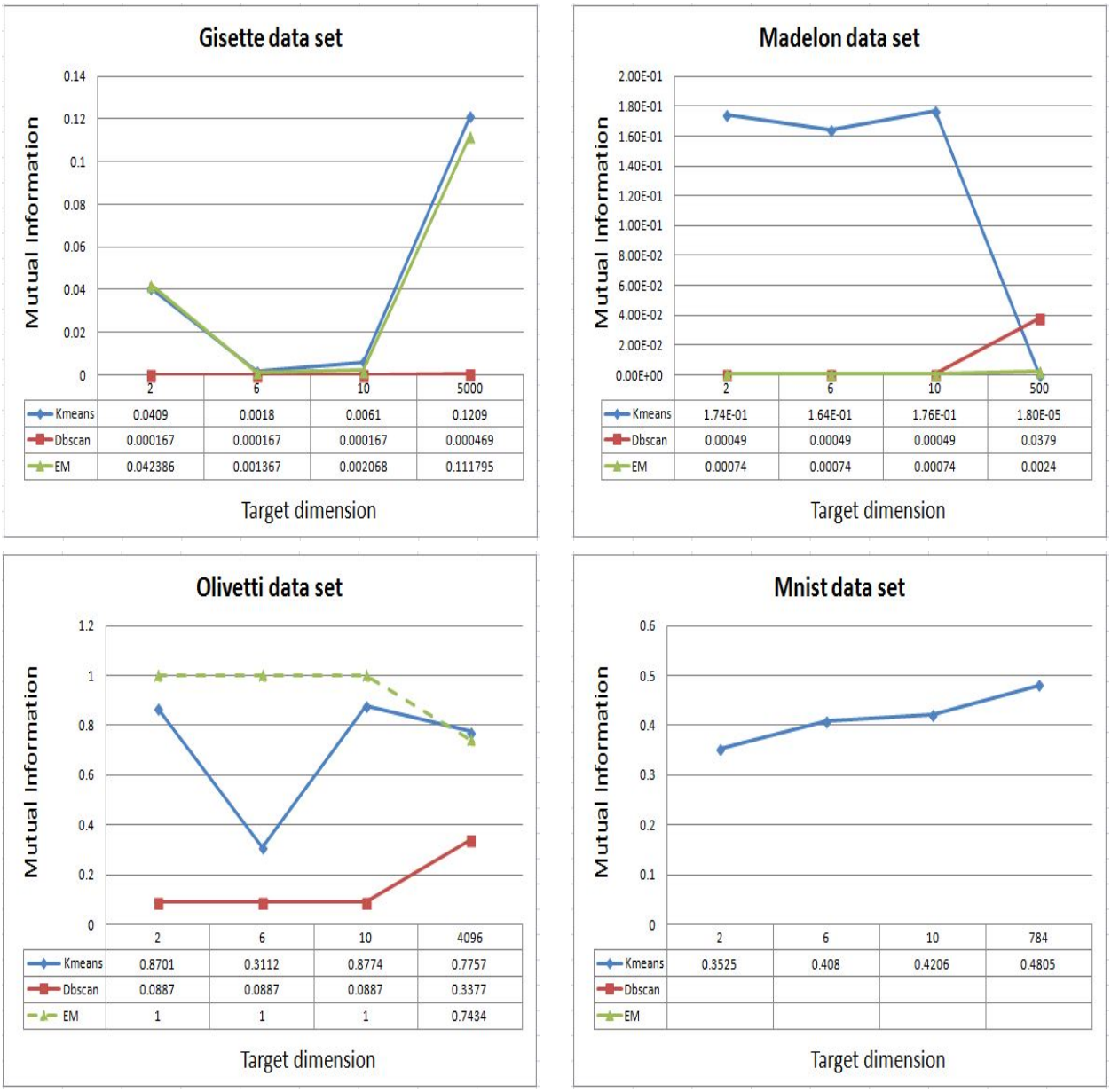

Figure 6: MI of datasets in original and reduced dimension space when reduced by LDA

The MI diagrams for four datasets when reduced by PCA is shown in figure 7.

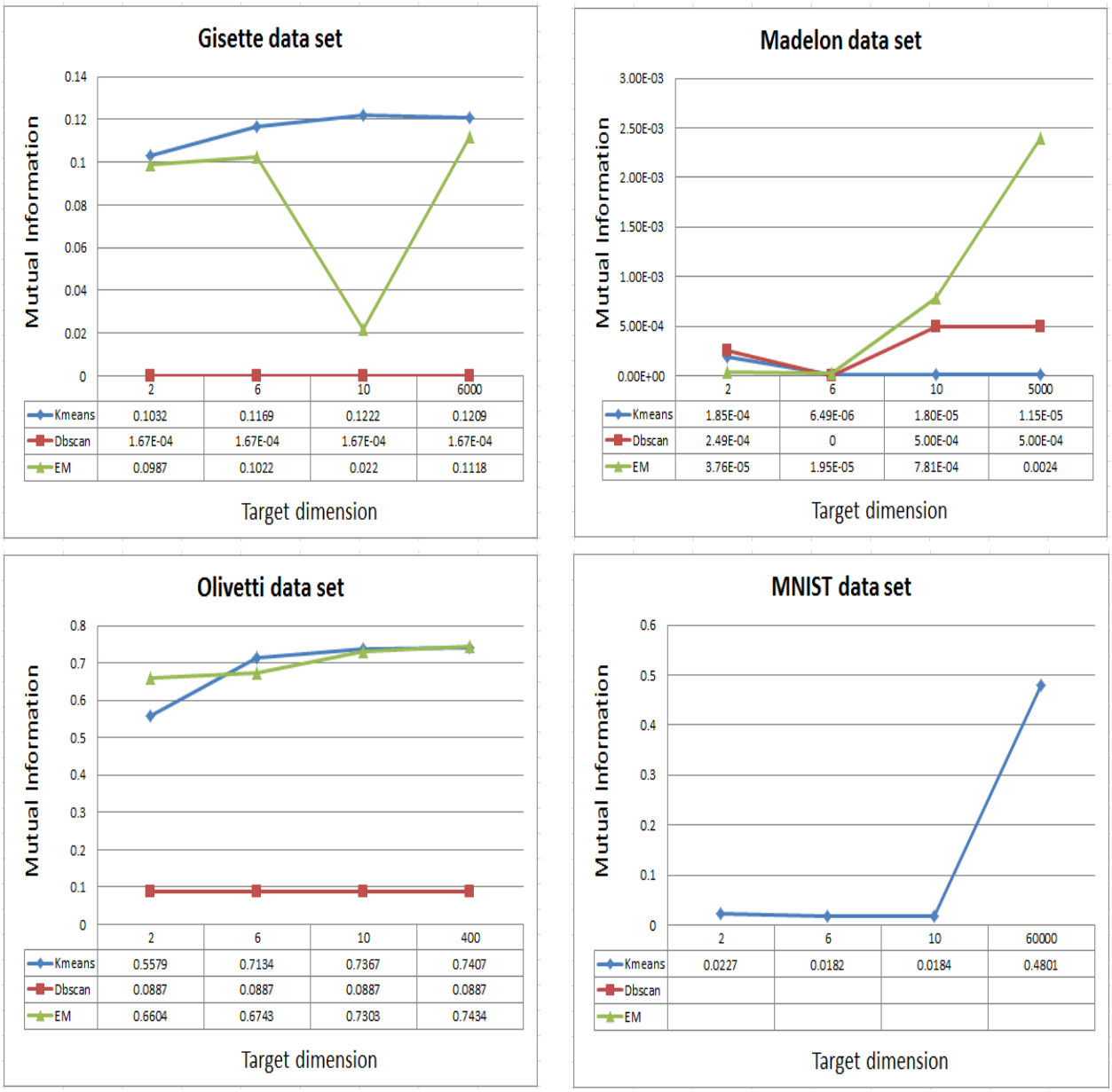

Figure 7: MI of datasets in original and reduced dimension space when reduced by PCA

6.3 Rand Index

The rand index diagrams for four datasets when reduced by LPI is shown in figure 8.

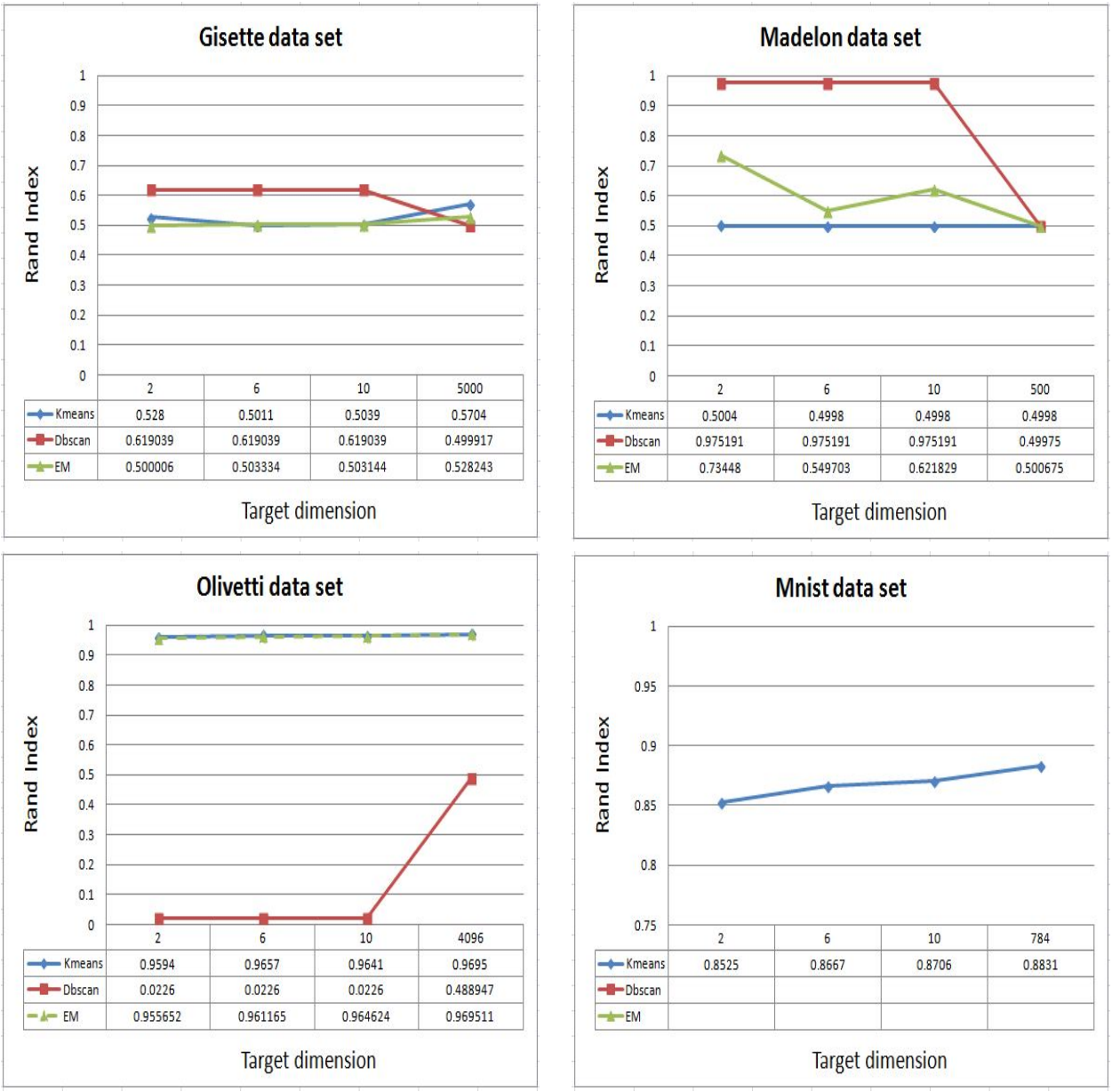

Figure 8: RI of datasets in original and reduced dimension space when reduced by LPI

The RI diagrams for four datasets when reduced by LDA is shown in figure 9.

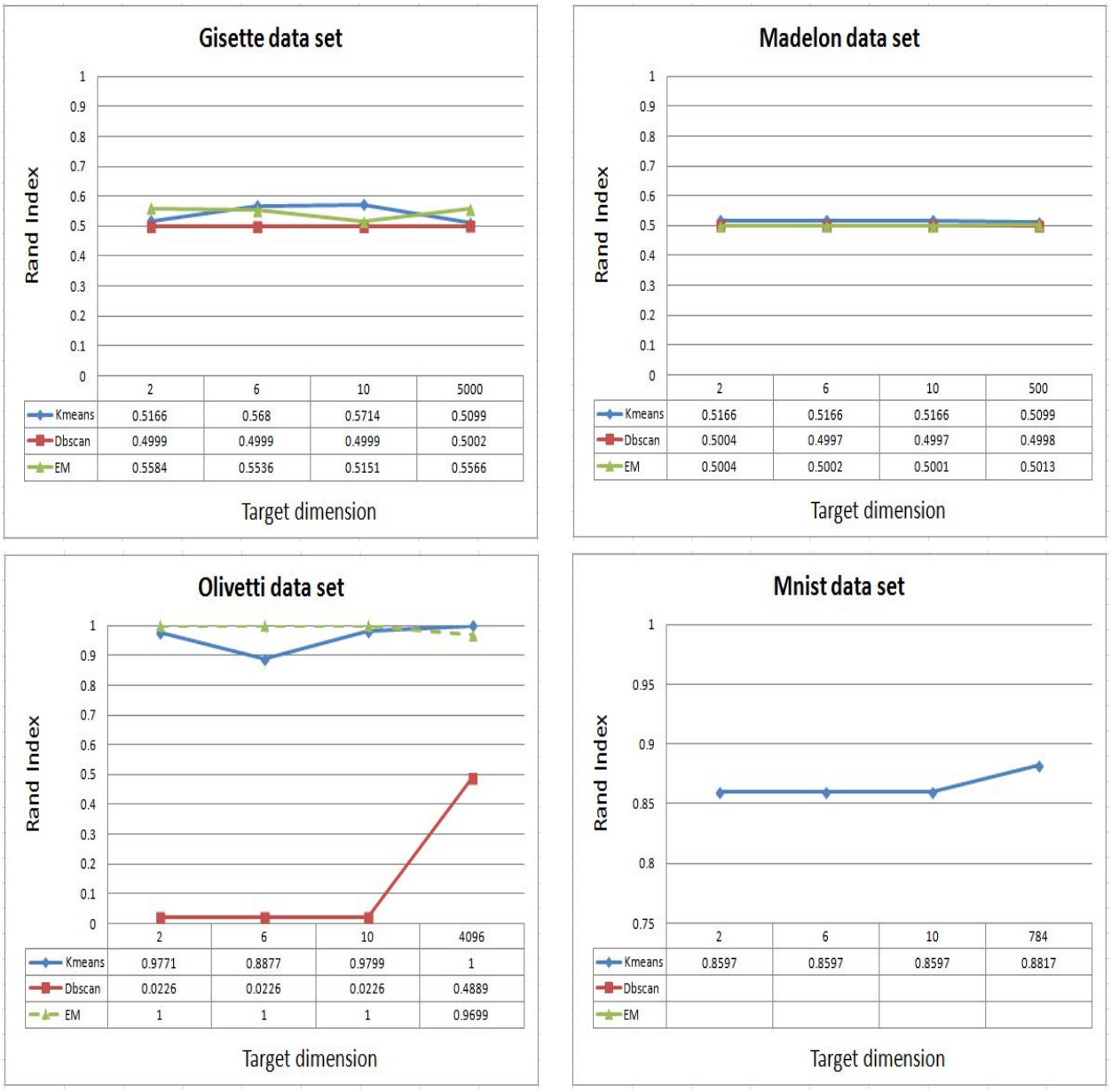

Figure 9: RI of datasets in original and reduced dimension space when reduced by LDA

The RI diagrams for four datasets when reduced by PCA is shown in figure 10.

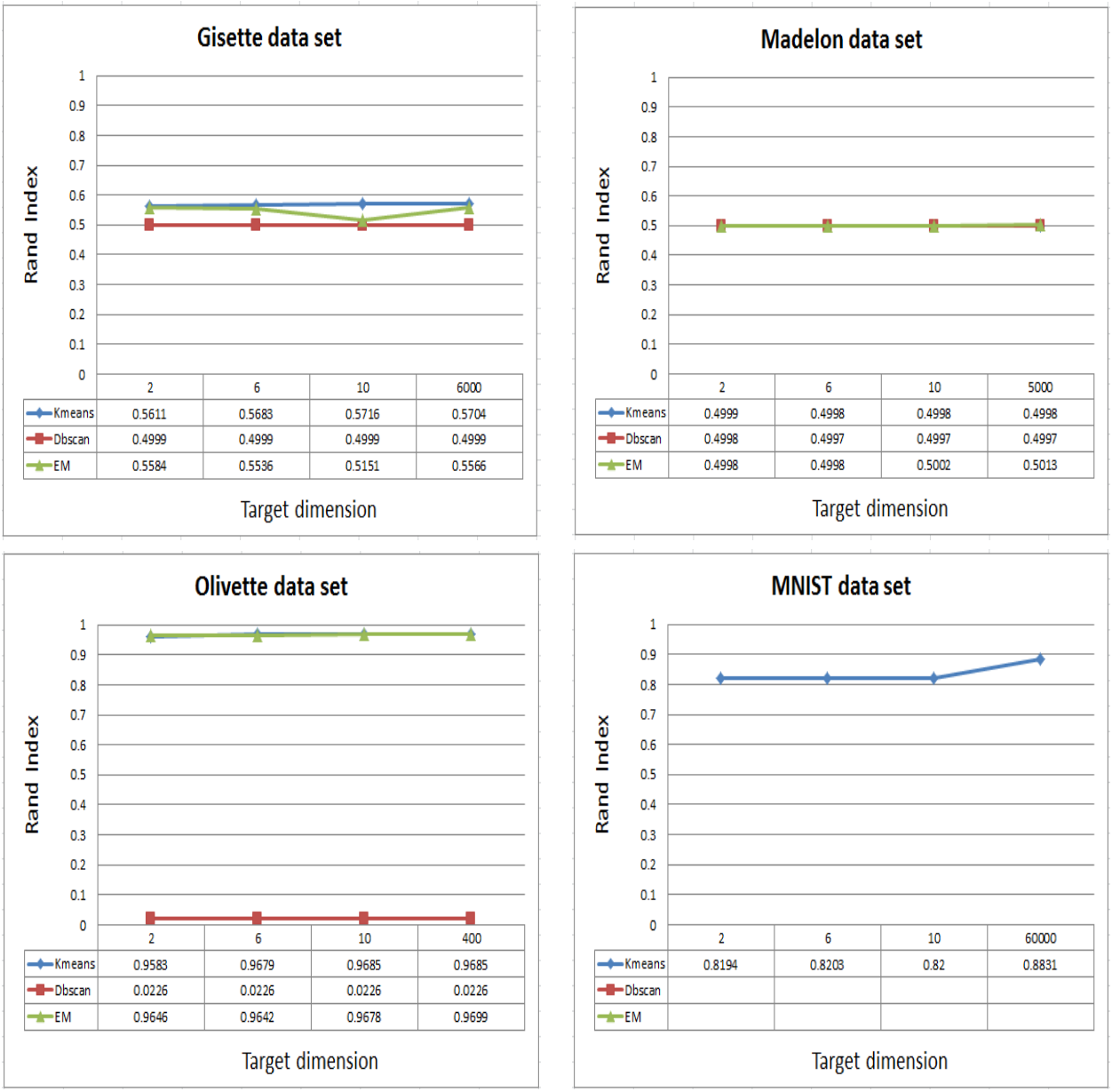

Figure 10: RI of datasets in original and reduced dimension space when reduced by PCA

7 Analysis

We performed various tests on the original data sets given to determine what kind of distribution might be present, the number of possible outliers (Grubbs and Mahalanobis) and repetition of data. We also computed standard deviation to see the data spread. For the outlier detection, our approach is not accurate., because as the degrees of freedom increase, we observe similar values for Mahalanobis Distance metric. It is still a decent approximation. We observed that Gisette and Madelon had a good amount of repetition. In addition Madelon had larger number of outliers compared to other data sets. Standard deviation is very high in gisette data set, while madelon and olivetti have values close to each other.

The Figure 11 shows standard deviation of datasets in original space and reduced dimension space.

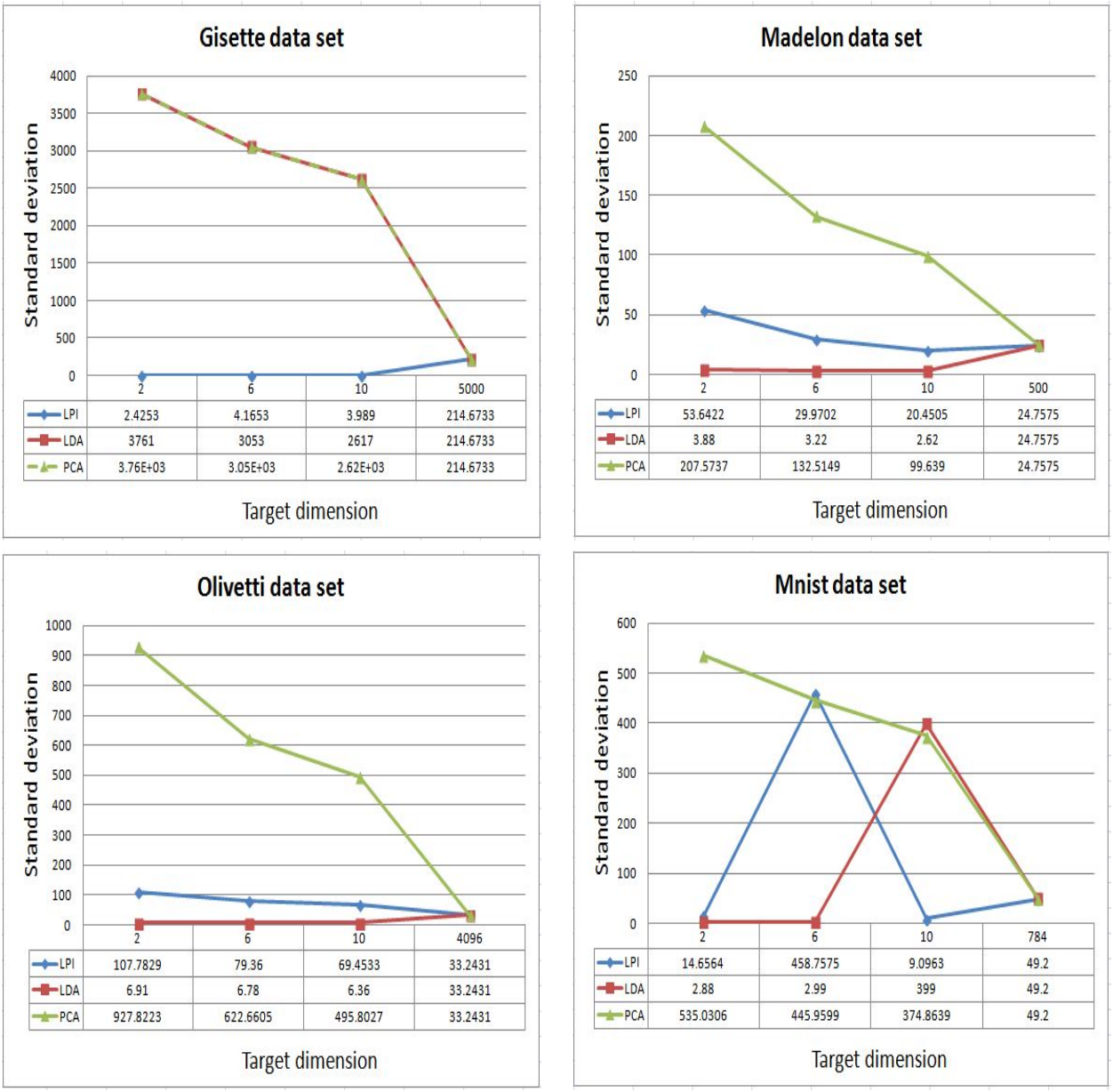

Figure 11: Standard deviation of datasets in original and reduced dimension space

We see that standard deviation in the reduced dimensions for madelon, ollivetti and mnist decrease while data is reduced by LDA and LPI, while for gisette it increases by a very large amount. As we see when data is reduced by PCA, standard deviation increases in lower dimension and that is due to the way PCA works, it reduces data in a way to preserve the maximum variance in it.

EM clustering is dependent upon how quickly points can coverge. Also we have higher noise in n-dimensions compared to reduced dimensions. As we are using K-Medoid EM Clustering algorithm, we think that noise averages out at a much higher scale in n-dimensions v/s reduced dimensions. This is validated when looking at the output of the Ollivetti and Madelon Data sets (figure 3). This data sets have lowest accuracy for EM in n-dimensions compared to their reduced dimensions. We also see that standard deviation decreases when we do dimension reduction on Ollivetti and Madelon and since we get better results with clustering the data distribution coming from DR techniques is better for the clustering algorithms. DR technique overcomes the sparsity of data present by some amount.

Ollivetti performs the best for LDA+EM as since it has least amount of repetition and outliers, the difference between within-class matrix is less and the projected means are far from each other.

We also observed that the number of clusters plays a significant role in the final output. On running the algorithms EM and K-means , we reduced the number of clusters than the given value as well as from increased them. In both cases we saw results deteroiating.

8 LDA Analysis

LDA + Kmeans works the best compared to LDA+ DBSCAN and LDA+EM. With LDA we restrict the space to be a linear combination of original dimensions, hence in this sub-space the clusters are well-seperated.

Consider a set of input data vectors $X = (x_1....x_i)$ in some high dimensional space. We know that kmeans clustering is to minimize the clustering objective function min $J_k, J_k = \sum_k \sum_{i \in C_k} ||x_i - m||^2$.

In LDA, total Scatter is $S_t = \sum_{i=1}^{n} x_i x_i^T$ between Class is Scatter $S_b = \sum_{k=1}^{n} n_k m_k m_k^T$ and within Class Scatter is $S_w = \sum_k \sum_{iC_k} (x_i - m_k)(x_i - m_k)^T$, and we know that $S_t = S_w + S_b$.

The objective function for Kmeans is $J_k = T_r \times S_w$ which T_r is the trace of the data matrix. We can write it as , $J_k = T_r \times (S_t - S_b)$

We can see that Kmeans clustering is also minimizing the within class scatter matrix S_w or maximizes the between class scatter matrix S_b.

We can also express LDA as, $max \frac{U^T S_b U}{U^T S_w U}$ which can also be expressed as min $T_r U^T S_w U$ and max $T_r U^T S_b U$.

We see both LDA and Kmeans have a similar objective function where we minimize the within-class scatter or maximize the between class scatter. This seems like we are doing the maximizing operation twice in the LDA + Kmeans process which helps in getting better results.

DBSCAN:Failed Approach

DBSCAN needs two parameters, minimum number of points and the radius for the cluster. In DBSCAN we form the cluster largely around noise points as border and core points which can go inside. So instead of looking for core points we started looking for noise points. We decided to try error-correcting codes to determing what can be the minimum number of points that can be passed and the distance metric for it. We used Hamming code to determine the minimum number of points. Sadly it did not work out after multiple runs :(

Similar to K-means explanation, in EM clustering the gaussian mixtures retain identical model in low-dimensional sub-space. EM fits a density functional form and in lower dimensional space we have smoother density which reduces our chances of converging in lots of iterations. This is where the LDA technique works well in EM as they have similar objective function.The irrelevant dimensions are removed by LDA and whichever are kept follows the Gaussian mixute functional form.

We see in figure 6 that Normalized Mutual Information is significantly less for all datasets except Ollivetti. As NMI is the comparison of number of points in a cluster before and after dimension reduction, we think that due to noise the important features are hidden with noise data and we are not able to see them. By performing DR techniques, we see that we are able to find this hidden features and using to get close to correct output. This inherently reduces NMI. As we said before Ollivetti has the least amount of noise and hence gives good results for NMI too.

Rand Index is the percentage of decisions that are correct. This translates to picking correct points from incorrect clusters and putting them into right clusters, as well as picking the noise points and putting them into the clusters which it thinks are correct. This would correct points over false positives and false negatives. Again as the number of outliers are less in olivetti it performs best for Rand Index also, while for the other data sets perform average (figure 9).

We have seen MNIST have less noise to compared to other data sets which comes across in the result of Kmeans.

But if the data distribution is far from linear then the deviations selected by LDA would substantially be different than the optimal choice. Hence we think that an increment/adaptive approach is better suited with LDA and a clustering algorithm like K-Means.

We propose that we use K-means clustering before DR technique to generate set of labels for the DR technique and then perform K-means clustering on the DR output. This would be our adaptive which we think should work better based upon the explanation given in the LDA+Kmeans objective function correlation.

LDA also has an issue where the intermediate scatter matrices have to be non-singular, as singular matrices lead to infinite eigen values. To overcome this situation we propose an approach where we apply PCA first on the original data matrix and obtain eigen values which are relevant. After that we apply LDA on the reduced data matrix by PCA to get the results which can be passed as input to clustering algorithms. We see that this improves the results of all data sets for evaluation metrics. We think this happens because PCA discards irrelevant dimensions based on the eigen values and gives us results based upon variance of data. Running LDA on these PCA generated eigen vectors gives us more opportunity to learn about discriminant information. The question remains about how much dimensions should we reduce in initial PCA such that discriminant information is not lost before applying LDA. We took the approach of trial and error to see at

what dimensions results start getting worse and then moved on to parameter estimation technique called k fold cross-validation to determine initial number of dimensions for PCA. This also overcomes one of the problems with LDA where if the discriminatory information is not in the mean but rather in the variance of data it will perform very poorly. LDA will perform poorly naturally if PCA is not able to get the correct. This is under the restriction that we reduce PCA to the correct number of dimensions. We present the improved values by the combination of PCA,LDA approach (figure 12). We show values where it made a significant increase in the evaluation metrics. The rest had a minor increment.

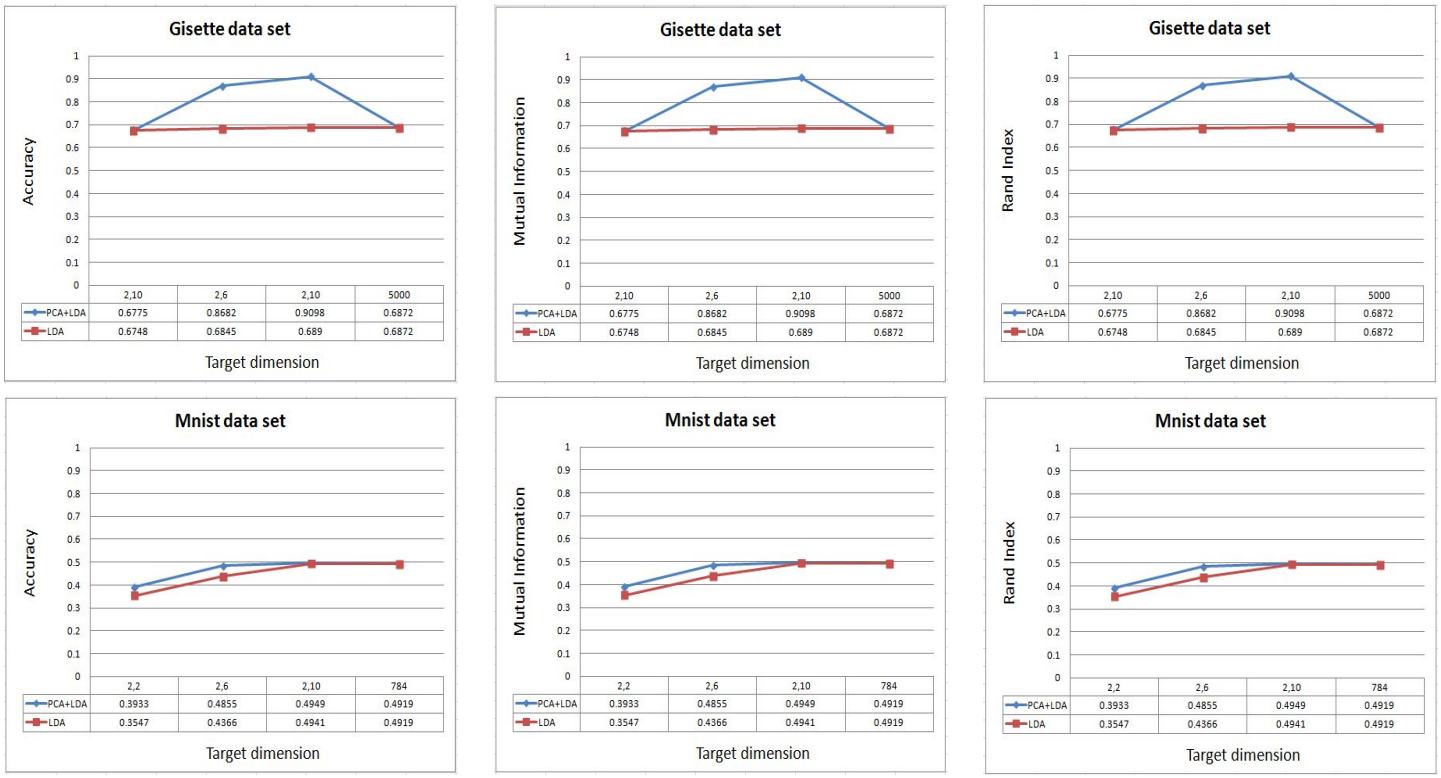

Figure 12: PCA+LDA approach on Gisette and Mnist data sets

In addition we tried another approach where instead of looking at the distance measures which most algorithms look at, we think we should look at the angle made by points. We can detect noise if a particular point has many other points which are located in similar directions, while it can be a feature if the points around it are in varying directions. We were able to get mixed results with this approach and since it was not implemented in the best possible manner, computationally we could not get it running to give us output esp. for mnist in a respectable amount of time.

9 PCA Analysis

Using PCA as a reduction algorithm did very little to change the accuracy of both DBScan and EM clustering's results (as it is shown in figure 4). DBScan in particular showed signs of improved speeds with hardly any change in results. EM Clustering showed some minor fluctuation in accuracy, but it surprisingly did better at lower dimensions (around 2) than at higher ones (around 10) on our dataset.

Kmeans seems to be the most susceptible to the affects of PCA, and showed fairly large decreases in accuracy as dimensions decreased, particularly in the mnist data set. It is possible that this reduction in accuracy could be minimized by either using a higher dimension size or a different distance metric than euclidean, but further study would be needed before drawing any particular conclusion.

If accuracy is a concern, PCA could possibly be a good choice to use in conjunction with DBScan, especially a set of good parameters for the clustering algorithm can be found. This has the ability to help improve the execution time of an otherwise costly algorithm without greatly reducing the algorithm's accuracy. PCA should generally be avoided however when using K-Means as it has the possibility to greatly loose accuracy as dimensions decrease.

Simply put, mutual information shows us how much our reduced datasets can tell us about our original one in regards to assigning a given point to a cluster(figure 7). Again, DBScan shows very little change in its MI score as dimensions are reduced using PCA. Surprisingly, K-means also shows a fairly stable MI score in

our datasets for Dimensions greater than 6 (although it showed a large drop in its score on the mnist database with any kind of reduction using PCA). EM clustering suffered the most of the three clustering algorithms, suggesting that much of the correlating information that it uses is quickly lost when doing PCA. Again, this suggests that the best algorithm to use PCA reduction with is DBScan, though a good set of parameters for the clustering algorithm is needed if it is to be effective.

Of the three analysis metrics that we used, RI showed the least amount of change across all of the clustering algorithms when using PCA (look at figure 10). There was some minor changes in gisette in conjunction with EM clustering and K-means on the mnist data set, but these small fluctuations mirror the bigger changes seen in the analysis of the algorithms' accuracy (see section 3.1). For this reason, there is very little to learn from this metric that we haven't already mentioned before.

Analysis: Summary

Probably the biggest advantage of using PCA for a reduction algorithm was it greatly improved the speed of our clustering algorithms such as DBScan (with the exception of the mnist data set). DBScan also seemed to maintain its correlating information, despite its dimensionality being greatly reduced. This coupled with the fact that PCA is simple to implement and not very resource intensive makes it an enticing approach for making datasets of extremely large dimensions easier and faster to process using resource intensive algorithms such as DBScan. Unfortunately, datasets with a large number of points will still be very computationally expensive when using DBScan as PCA can only reduce their dimensionality and not the number of points that need to be compared. If however a reduction algorithm is needed to be used along side DBScan, PCA seems to be a good choice.

10 LPI Analysis

As we can see in figure 2, LPI+K-means and LPI+EM works much better than LPI+Dbscan, however this performance is still far worse than performance of LDA and PCA (while the accuracy of LDA+K-means and PCA+K-means is around 0.7, LPI+K-means has an accuracy of about 0.5). Since we have redundancy in data as checked by the repetition tests, it is possible that manifold space selected by LPI might be based around this redundant points. Since it does not take class labels as input to do any kind of discrimination, it cannot distinguish between noise and data points. Apart from this we tried one more thing, where after reducing from n dimension to k dimension, the features that we have to should be removed from the original data set. This would point to if there are any other features present which are orthogonal to the features we found. If we found no features in the original data set (the data set now does not have the features which we found by LPI), then this would be point to best feature found. We tried doing this but were not able to come up with a sound solution.

One of drawback of conventional LPI is that if it is implemented by solving the minimum eigenvalue problem, the minimum eigenvalue solution is not always optimal for preserving the local structure. There are two reasons for this. The first is that if there are zero eigenvalues, conventional LPI will take as transforming axes the eigenvectors correspondig to the zero eigenvalues of th generalized eigen equation. As a resutl, after conventional LPI transforms samples into a new space using these transforming axes, a sample statistically will have the same representation as its neighbours. This is not how locality preserving projection works. The goal of LPI is not to make samples have the same representation but is to preserve the neigbour relationships between samples. The second reason is that the classifier can not correclty classify samples when conventional LPI is implemented in the unsupervised case, since two neighbor samples from two different classes might have the same representation in the new space.

As clustering algorithms K-Means and K-Medoid EM work on finding clusters based upon euclidean distance measure, the sub-space that is preserved by LPI does not have the global structural information which can be efficient for clustering. We verified the lack of class labels in LPI approach by doing a simple experiment. We ran LDA on a dataset and then applied LPI on the data set reduced from LDA. This drastically improved the clustering in terms of speed as well as the evaluation metrics. We think in cases where there are lot of non-orthogonal features, a pre-processing step with DR technique having class labels followed by LPI will perform much better. We would like this to be an adaptive approach of LDA+LPI. Results of this approach are shown in figure 13. Since Gisette and Madelon have the singularity problem, were could not apply this approach on them.

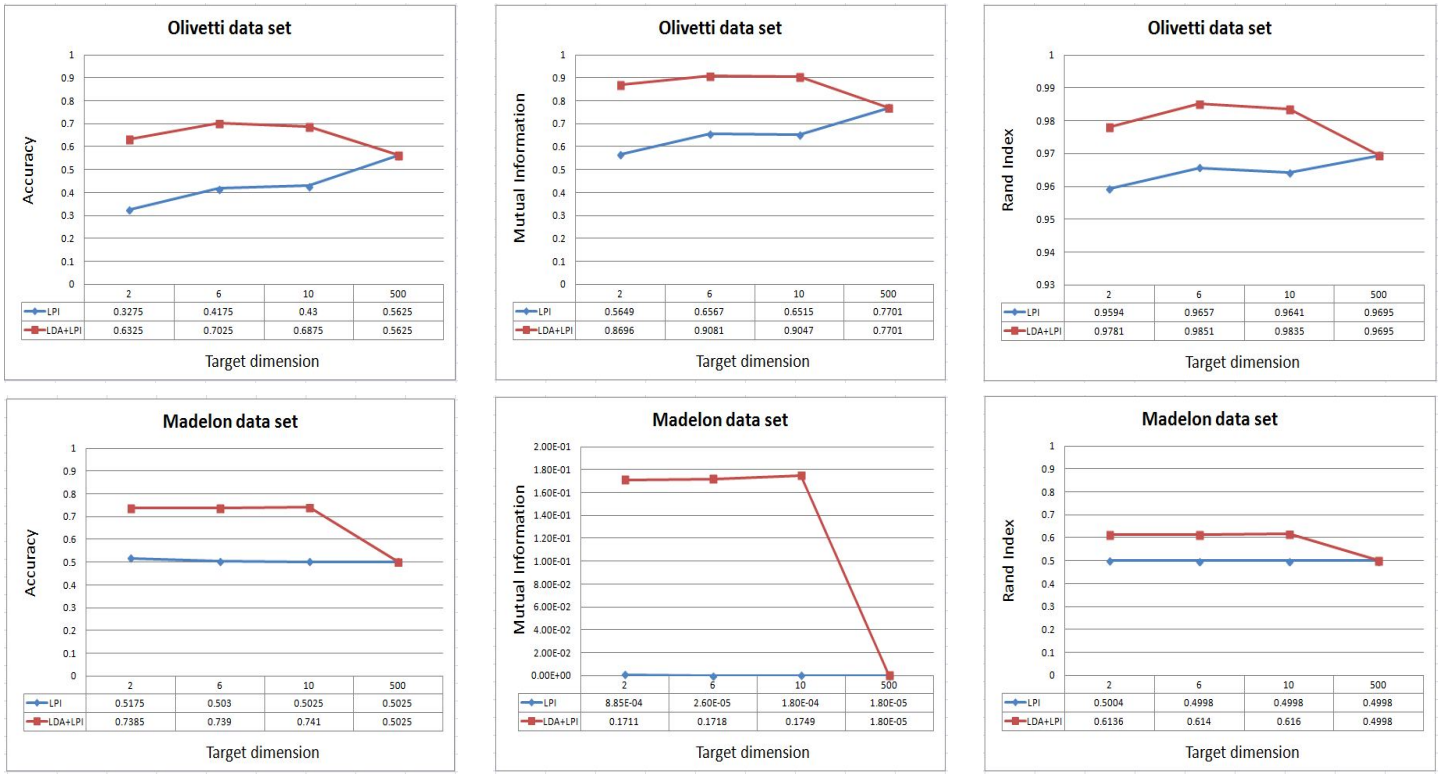

Figure 13: LDA+LPI algorithm on Madelon and Olivetti data sets

11 Conclusion

Clustering on the original data sets is not an effective approach because there is no thought process in the clustering algorithms that we looked at in terms of finding structure in data. Since structure in data is not expressed in terms of any possible distance metric, clustering usually performs worse compared to the DR+Clustering output. DR techniques help in finding the relevant sub-space over which we should do clustering. If a particular distance metric is preserved in DR process, then that same distance metric will also be very effective while performing clustering.

References

[1] Cai, D.,He, X., Han, J.:Document clustering using locality preserving indexing. In: IEEE Transactions on Knowledge and Data Engineering,17, pp. 16241637 (2005)

[2] Ding, C.: A Similarity-based Probability Model for Latent Semantic Indexing. Proc. of 22nd ACM SIGIR Conference, pp. 59-65 (1999)

[3] Duda, R.O., Hart, P.E., Stork, D.G.:Pattern Classification. Wiley-Interscience, Hohn Wiley and Sons, 2nd edition edition (1995)

[4] Ester, M., Kriegel, H., Sander, J., Xu, X.:A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226231 (1996)

[5] Xu, W., Liu, X., Gong,Y.: Document clustering based on non-negative matrix factorization. In: 3rd International Conference on Research and Development in Information Retreival, pp. 267-273, Toronto, Canada (2003)

[6] Zhang, Z., Zha, H.:Structure and perturbation analysis of truncated SVD for column-partitioned matrices. Matrix Analysis and Applications, 22, pp. 12451262 (2001)

[7] Dhillon, I.S., Modha, D.S.:Concept decompositions for large sparse text data using clustering. Machine learning. 42, pp. 143175 (2001)

[8] Gao, J., Zhang, J.:Clustered SVD strategies in latent semantic indexing. Information Processing and Management. 41, pp. 10511063 (2005)

[9] Zeimpekis, D., Gallopoulos, E.:ClSI: A flexible approximation scheme from clustered term-document matrices. In: SIAM Data Mining Conference, pp. 631635, Newport Beach, California (2005)