Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach

Xiaoli Zhang Fern XZ@ECN.PURDUE.EDU
Carla E. Brodley BRODLEY@ECN.PURDUE.EDU
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907

Abstract

We investigate how random projection can best be used for clustering high dimensional data. Random projection has been shown to have promising theoretical properties. In practice, however, we find that it results in highly unstable clustering performance. Our solution is to use random projection in a cluster ensemble approach. Empirical results show that the proposed approach achieves better and more robust clustering performance compared to not only single runs of random projection/clustering but also clustering with PCA, a traditional data reduction method for high dimensional data. To gain insights into the performance improvement obtained by our ensemble method, we analyze and identify the influence of the quality and the diversity of the individual clustering solutions on the final ensemble performance.

1. Introduction

High dimensionality poses two challenges for unsupervised learning algorithms. First the presence of irrelevant and noisy features can mislead the clustering algorithm. Second, in high dimensions data may be sparse (the curse of dimensionality), making it difficult for an algorithm to find any structure in the data. To ameliorate these problems, two basic approaches to reducing the dimensionality have been investigated: feature subset selection (Agrawal et al., 1998; Dy & Brodley, 2000) and feature transformations, which project high dimensional data onto "interesting" subspaces (Fukunaga, 1990; Chakrabarti et al., 2002). For example, principle component analysis (PCA), chooses the projection that best preserves the variance of the data.

Noise & sparse makes it hard to find structure in high dim data.

In this paper we investigate how a relatively new transformation method, random projection (Papadimitriou et al., 1998; Kaski, 1998; Achlioptas, 2001; Bingham & Mannila, 2001), can best be used to improve the clustering result for high dimensional data. Our motivation for exploring random projection is twofold. First, it is a general data reduction technique. In contrast to other methods, such as PCA, it does not use any defined "interestingness" criterion to "optimize" the projection. For a given data set, it may be hard to select the correct "interestingness" criterion. Second, random projection has been shown to have special promise for high dimensional data clustering. In 1984, Diaconis and Freedman showed that various high-dimensional distributions look more Gaussian when randomly projected onto a low-dimensional subspace. Recently, Dasgupta (2000) showed that random projection can change the shape of highly eccentric clusters to be more spherical. These results suggest that random projection combined with EM clustering (of Gaussian mixtures) may be well suited to finding structure in high dimensional data.

Don't need some criterion to cluster over.

However, the drawback of random projection is that it is highly unstable – different random projections may lead to radically different clustering results. This instability led us to investigate a novel instantiation of the cluster ensemble framework (Strehl & Ghosh, 2002) based on random projections. In our framework, a single run of clustering consists of applying random projection to the high dimensional data and clustering the reduced data using EM. Multiple runs of clusterings are performed and the results are aggregated to form an $n \times n$ similarity matrix, where n is the number of instances. An agglomerative clustering algorithm is then applied to the matrix to produce the final clusters. Experimental results on three data sets are presented and they show that the proposed cluster ensemble approach achieves better clustering performance than not only individual runs of random projection/clustering but also EM clustering with PCA data reduction. We also demonstrate that both the quality and the diversity of individual clustering solutions have strong impact on the resulting ensemble performance.

Problem w/ random projection is instability so ensemble method is a natural solution.

1 Run: Use random projection & EM for clustering.
Agg results of n runs into nxn sim. matrix,
then cluster the matrix for final result.

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

Shows the results of clustering are unstable,

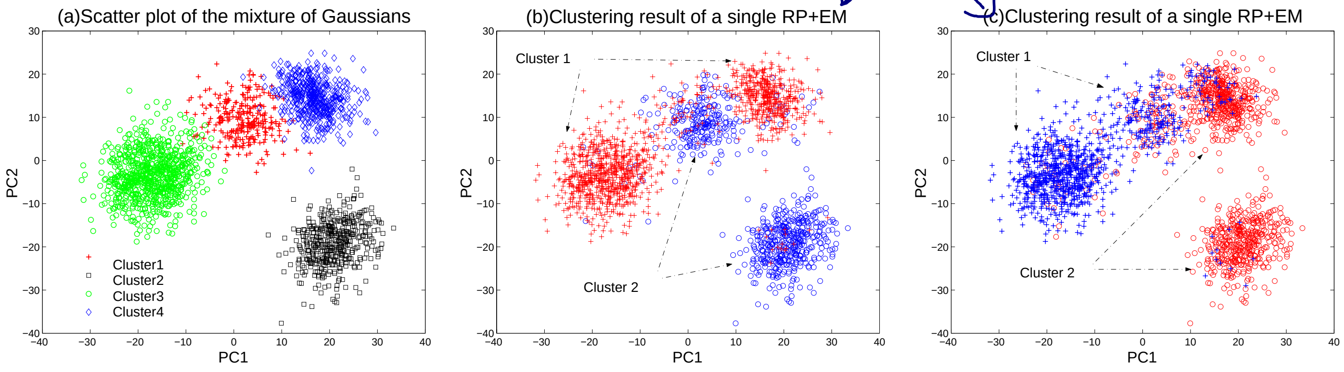

Figure 1. (a)The original clusters; (b) and (c) Two representative clustering results generated by RP+EM

2. Random Projection and the Cluster Ensemble Approach

In this section we first illustrate that although a single run of random projection may lead to a less than satisfactory clustering result, it is possible to uncover the natural structure in the data by combining the results of multiple runs. We then present a new approach to clustering high dimensional data that first combines the information of multiple clustering runs to form a "similarity" matrix and then applies an agglomerative clustering algorithm to produce a final set of clusters.

2.1. A Single Random Projection

R, dxd'

X, nxd

XR is the projection.

A random projection from d dimensions to d' dimensions is a linear transformation represented by a $d \times d'$ matrix R, which is generated by first setting each entry of the matrix to a value drawn from an i.i.d N(0,1) distribution and then normalizing the columns to unit length. Given a d-dimensional data set represented as an $n \times d$ matrix X, where n is the number of data points in X, the mapping $X \times R$ results in a reduced-dimension data set X'.

We applied random projection to a synthetic data set that consists of two thousand data points forming four Gaussian clusters in a fifty-dimensional space. For this data, we chose a random projection that reduces the data to five dimensions[1] producing data set X'. The EM algorithm was then applied to cluster X'. Although the real number of clusters is known, we used the Bayesian Information Criterion (BIC) (Fraley & Raftery, 1998) to determine the number of clusters k because in most real applications k is unknown. In addition, the natural number of clusters may vary in different subspaces (Dy & Brodley, 2000).

Using the first two principle components, Figure 1 plots the original clusters and two representative examples of the clusters formed by random projection with EM clustering (RP+EM). The number of clusters chosen by RP+EM varied from run to run (in the figure both RP+EM results have two clusters) and occasionally RP+EM found all four clusters. The scatter plot of the RP+EM results suggest that random projection may distort the underlying structure of the data and result in unstable clustering performance. To some degree this contradicts Dasgupta's results (2000), which show that random projection preserves the separation among Gaussian clusters. However, Dasgupta's results were averaged across many runs and when the projected dimension was small, the results tended to have large variance. An alternative explanation is that for this data set, random projection needs more than five dimensions to preserve the structure. There have been results (Achlioptas, 2001) about the required dimensionality for a random projection to effectively preserve *distance*. However, to our knowledge it is still an open question how to choose the dimensionality for a random projection in order to preserve *separation among clusters* in general clustering applications.

On viewing these results, the first conclusion one makes is that for this data set an individual run of RP+EM (with dimensionality of five) produces suboptimal results. Further inspection reveals that *different runs may uncover different parts of the structure in the data that complement one another.* From Figure 1(b) and (c) we observe that each run uncovered some partial, but different, structure in the data. This suggests that the combination of multiple runs of RP+EM may lead to better results – we may be able to reveal the true structure of the data even without the knowledge of how to choose a proper dimensionality to preserve the original cluster structure.

Complimentary errors makes the ensemble method work.

[1] Our choice of five was based on the observation that with the first five principle components EM can recover the four clusters with over 95% accuracy.

Agg method works, P_{ij} is sig. greater when i, j are from same cluster.

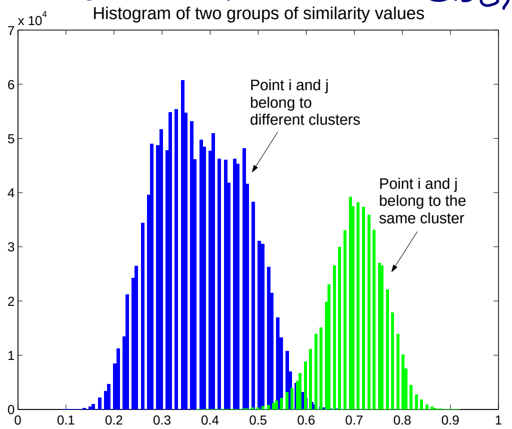

Figure 2. The histogram of two groups of similarity values. The left group consists of the P_{ij} values for pairs of points from different clusters. The right group consists of the P_{ij} values for pairs of points from the same cluster.

2.2. Multiple Random Projections

We combine the results of multiple runs of RP+EM with a two-step process. First, we aggregate the clustering results into a matrix that measures the "similarity" between each pair of data points. Then an agglomerative clustering algorithm is applied to produce the final clusters. We discuss these steps below.

Aggregating multiple clustering results: For each run of random projection, EM generates a probabilistic model θ of a mixture of k Gaussians[2] in the projected d'-dimensional space. For data point i, the soft clustering results $P(l|i,\theta), l = 1,\ldots,k$ are given, representing the probability that the point belongs to each cluster under the model θ. We define P^θ_{ij} as the probability of data point i and j belonging to the same cluster under model θ and it can be calculated as:

$$P^\theta_{ij} = \sum_{l=1}^{k} P(l|i,\theta) \times P(l|j,\theta)$$

To aggregate multiple clustering results, the values of P^θ_{ij} are averaged across n runs to obtain P_{ij}, an "estimate" of the probability that data point i and j belong to the same cluster. This forms a "similarity" matrix. We expect the P_{ij} values to be large when data point i and j are from the same natural cluster and small otherwise. To test our conjecture, we performed thirty[3] runs of RP+EM on the synthetic data set and separated the aggregated P_{ij} values into two groups based on if data point i and j are from the same cluster. Figure 2 shows the histograms of both groups of P_{ij} values. It can be seen that the distributions of the two groups have different means and little overlap, which supports our conjecture. Note that if the two distributions are completely separated the true clustering structure can be easily recovered by thresholding P_{ij}'s.

Agg Method: For each run of RP+EM, compute $P^{\theta}_{ij} = \sum_{l=1} P(l|i) P(l|j)$ & then average across runs.

Table 1. The basic agglomerative clustering algorithm

```
Inputs:   $P$ is a $n \times n$ similarity matrix,
          $k$ is a desired number of clusters.
Output:   a partition of $n$ points into $k$ clusters.

Procedure: An Agglomerative Clustering Algorithm
$l = n$.
For $i = 1$ to $n$
    Let $c_i = \{x_i\}$ for $i = 1, \ldots, n$
Repeat
    Find the most similar pair of clusters based
      on $P$, say $c_i$ and $c_j$.
    Merge $c_i$ and $c_j$ and decrement $l$ by one
Until $l \leq k$
Return all nonempty clusters
```

Producing the final clusters: To produce the final clusters from the aggregated "similarity" matrix P, we apply an agglomerative clustering procedure whose basic steps are described in Table 1.

In implementing the agglomerative algorithm, we need to define the similarity between two clusters and determine the proper number of clusters for a given data set. In our implementation, we define similarity as:

$$sim(c_i, c_j) = \min_{x_i \in c_i, x_j \in c_j} P_{ij}$$

This is equivalent to the complete-link distance-based agglomerative clustering algorithm (Duda & Hart, 1973). We chose this definition to ensure that when two points have very small "similarity" value (i.e., small possibility of belonging together according to P) the algorithm will not group them together.

So this sim formula makes sure that very dissimilar points are never grouped together.

In the experiments we observe that some data points are not similar to any other data points. Intuitively the decision of merging should not be based on these points because they can be the "outliers" of the data set. To avoid the impact of such points, we remove them during the merging process and assign them to the formed clusters afterward. Specifically, we calculate the maximum similarity between data point i to the other data points as $P_{max}(i) = \max_{j=1}^{n} P_{ij}$, where $j \neq i$. In the merging process, we discard 10% of the data points with the smallest P_{max} values. After merging we then assign these points to their most similar clusters, where the similarity between a data point i and a cluster c_k is defined[4] as : $\frac{1}{\|c_k\|} \sum_{x_j \in c_k} P_{ij}$.

Don't consider outlier points in merging.

[2] Note that k may vary from run to run.

[3] This is an arbitrary choice.

[4] We chose to use the average instead of the minimum to define the similarity here because the minimum measure will be biased toward small clusters.

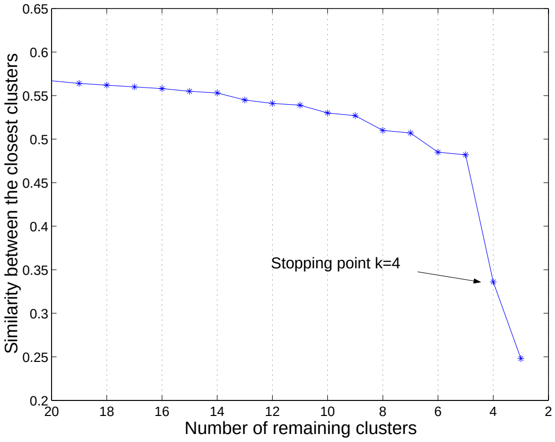

Figure 3. The similarity between the closest clusters in the merging process

Pretty nice way to pick k since we will consider all possible values.

To decide the cluster number k, we cannot apply commonly used techniques such as BIC because our method does not generate any description or model for the clusters. To solve this problem, we propose to continue merging until only a single cluster remains and at each step plot the similarity between the two clusters selected for merging. Figure 3 shows the plot of these similarity values for the synthetic data set described in Section 2.1. We observe a sudden drop of the similarity when the algorithm tries to merge two real clusters. In the experiments on other data sets, we also observed similar trends from the plots. This suggests that we can use the occurrence of a sudden similarity drop as a heuristic to determine k.

Using the steps described above, we combined the results of thirty runs of RP+EM and successfully recovered all four clusters from the synthetic data set with 100% accuracy.

In summary, our method consists of three steps: 1) generate multiple clustering results using RP+EM, 2) aggregate the results to form a "similarity" matrix, and 3) produce final clusters based on the matrix. Note that there are many different choices we can make for each of the three steps. Section 4 discusses the desirable properties for the first step and a possible strategy for improving our approach.

3. Experimental Results

Our experiments are designed to demonstrate: 1) the performance gain of multiple random projections over a single random projection, and 2) that our proposed ensemble method outperforms PCA, a traditional approach to dimensionality reduction for clustering.

Table 2. Summary of the data sets

Data set	HRCT	CHART	EOS
#Inst.	1545	600	2398
#Class	8	6	8
Org. Dim.	183	60	20
RP Dim.	10	5	5
PCA Dim.	30	5	5

3.1. Data Sets and Parameter Settings

We chose two data sets with a relatively high dimensionality compared to the number of instances (HRCT and CHART) and one data set with a more traditional ratio (EOS). Table 2 summarizes the data set characteristics and our choice for the number of dimensions that we used for random projection and PCA. HRCT is a high resolution computed tomography lung image data set with eight classes (Dy et al., 1999). CHART is a data set of synthetically generated control chart time series with six different types of control charts (Hettich & Bay, 1999). EOS is an eight-class land cover classification data set. Although the class labels are available for all three data sets, they are discarded in the clustering process and only used during evaluation.

We selected the dimensionality of PCA for each data set by requiring that 85% of the data variance be preserved. For the CHART and EOS data sets, we set the dimensionality for RP to be the same as PCA in order to have a more direct comparison. However, for the HRCT data set, we chose a much smaller number than that chosen for PCA for computation time reasons.

3.2. Evaluation Criteria

Evaluating clustering results is a nontrivial task. Because our method does not generate any model or description for the final clusters, internal criteria such as log-likelihood and scatter separability (Fukunaga, 1990) can not be applied.[5] Because our data sets are labeled, we can assess the cluster quality by using measures such as conditional entropy and normalized mutual information (Strehl & Ghosh, 2002). We chose to report results for both criteria because as we explain below entropy is biased toward a large number of clusters and normalized mutual information under some conditions is biased toward solutions that have the same number of clusters as there are classes.

Conditional Entropy (CE): Conditional entropy measures the uncertainty of the class labels given a

[5] Such internal criteria would require calculation either in the original or a transformed feature space. Because we have multiple transformations there is no single "feature space."

clustering solution. Given m classes and k clusters, for a particular class $i \in [1..m]$ and cluster $j \in [1..k]$, we first compute p_{ij}, which is the probability that a member of cluster j belongs to class i. The entropy of the class labels conditioned on a particular cluster j is calculated as: $E_j = -\sum_{i=1}^{m} p_{ij} log(p_{ij})$. The conditional entropy (CE) is then defined as: $CE = \sum_{j=1}^{k} \frac{n_j * E_j}{n}$, where n_j is the size of cluster j and n is the total number of instances.

We would like to minimize CE. Its value is 0 when each cluster found contains instances from only a single class. Note that we can also obtain a value of 0 if each cluster contains a single instance. Therefore it is clear that this criterion is biased toward larger values of k because the probability of each cluster containing instances from a single class increases as k increases. Because of this bias, we use CE only when comparing two clustering results with the same value of k.

Normalized Mutual Information(NMI): For a detailed description of NMI see (Strehl & Ghosh, 2002). Let X be a r.v. representing the distribution of *class* labels $[1..m]$ and Y be a r.v. representing the distribution of *cluster* labels $[1..k]$. To calculate NMI between r.v.'s X and Y, we first compute the mutual information between X and Y as $MI = \sum_{i,j} p_{ij} log(\frac{p_{ij}}{p_i p_j})$ where p_{ij} is defined as above, p_i is the probability of class i, and p_j is the probability of cluster j. Mutual information measures the shared information between X and Y. Note that its value is not bounded by the same constant for all data sets. NMI normalizes it onto the range [0,1] by: $NMI = \frac{MI}{\sqrt{H(X)H(Y)}}$ where $H(X)$ and $H(Y)$ denote the entropy of X and Y.

NMI attains the optimal value of 1.0 when there is a one to one mapping between the clusters and the classes (i.e., each cluster contains one class and $k = m$). Unlike CE and other criteria such as the RAND index (Rand, 1971), NMI is not biased by large k (Strehl & Ghosh, 2002). However, note that if a class is multi-modal and our clustering solution correctly reflects this, the value of NMI will not be 1.0.

3.3. Comparison with RP+EM

Our method is built on the conjecture that combining multiple runs of RP+EM is better than a single run of RP+EM or PCA+EM. In this section we present the results of a comparison between our ensemble approach and individual runs of RP+EM.

We performed thirty runs of RP+EM to form a cluster ensemble for each data set. Note that we force each run of RP+EM to find a fixed number of clusters, which is set to be the number of classes. Although this is not necessary (or even desirable), it lets us remove the effect of k in evaluating the cluster results.

Table 3. Cluster ensemble versus single RP+EM

Data set		HRCT	CHART	EOS
NMI	Ens.	0.236	0.790	0.253
	Sin.	0.174±0.03	0.481±0.09	0.240±0.02
CE	Ens.	1.743	0.706	1.923
	Sin.	1.907±0.07	1.410±0.22	1.960±0.04

Table 3 reports the CE and NMI values of both the ensemble and individual runs of RP+EM. The values reported for RP+EM are averaged across the thirty runs. From the table we see that the ensemble improves the clustering results over its components for all three data sets (recall CE is to be minimized and NMI is to be maximized). The difference in performance is the least for EOS, which we conjecture is due to less diversity in the individual clustering results for EOS. We explore this conjecture in Section 4.

3.4. Comparison with PCA+EM

We compare our approach to PCA with EM clustering (of Gaussian Mixtures) in two scenarios: 1) where we force both methods to produce the same number of clusters, and 2) where we allow each approach to adapt the cluster number to its clustering solution.

In our first experiment we require both methods to form the same number of clusters. This allows us to remove the influence of k on the evaluation criteria. We want to clarify that the individual runs of RP+EM still use BIC to search for k, because this will not bias the evaluation criteria and further because different runs naturally require different numbers of clusters (Dy & Brodley, 2000). Note that the "natural" number of clusters in the data may or may not equal to the number of classes in our labeled data sets, therefore we compare the two methods for five different values of k. The cluster ensembles are formed by thirty runs of RP+EM and the reported results are averaged across five runs of each method.

Table 4 shows CE and NMI for five different values of k. Recall that we want to maximize NMI and minimize CE and further that NMI ranges from 0 to 1 and CE ranges from 0 to $log_2(m)$, where m is the number of classes. For all three data sets, our method results in better performance as measured by both CE and NMI except for $k = 14$ for the CHART data set. We observe that the ensemble tends to have smaller variance than PCA+EM, particularly when k is large. From these results, we conclude that for these data sets 1) the ensemble method produces better clusters, and 2) it is more robust than PCA+EM.

Table 4. Results for the cluster ensemble versus PCA+EM

HRCT DATA SET

k		8	10	12	14	16
NMI	ENSEM.	0.233±0.015	0.244±0.008	0.256±0.006	0.267±0.007	0.279±0.007
	PCA	0.219±0.048	0.220±0.028	0.222±0.015	0.237±0.015	0.225±0.039
CE	ENSEM.	1.765±0.063	1.714±0.047	1.659±0.046	1.604±0.039	1.556±0.040
	PCA.	1.813±0.137	1.779±0.085	1.751±0.047	1.687±0.059	1.699±0.129

CHART DATA SET

k		6	8	10	12	14
NMI	ENSEM.	0.700±0.061	0.783±0.006	0.769±0.006	0.758±0.008	0.747±0.012
	PCA	0.654±0.081	0.665±0.066	0.691±0.065	0.697±0.075	0.738±0.032
CE	ENSEM.	0.947±0.251	0.675±0.009	0.552±0.063	0.540±0.063	0.526±0.047
	PCA.	1.028±0.249	0.852±0.229	0.711±0.176	0.563±0.220	0.375±0.074

EOS DATA SET

k		8	10	12	14	16
NMI	ENSEM.	0.282±0.015	0.276±0.008	0.279±0.006	0.280±0.007	0.280±0.007
	PCA	0.239±0.009	0.251±0.012	0.263±0.011	0.267±0.006	0.269±0.006
CE	ENSEM.	1.866±0.022	1.832±0.011	1.785±0.023	1.760±0.015	1.738±0.021
	PCA.	1.958±0.023	1.895±0.041	1.829±0.039	1.790±0.023	1.764±0.023

Table 5. Cluster ensemble versus PCA+EM with different *k* values

	PCA+EM		ENSEMBLE	
DATASET	NMI	K	NMI	K
HRCT	0.191 ±0.05	6.2	0.248 ±0.01	10
CHART	0.721 ±0.05	7.8	0.786 ±0.01	7.6
EOS	0.265 ±0.03	14.3	0.281 ±0.01	14.2

In our second comparison, we allow each method to determine its own cluster number, *k*. To decide *k*, the cluster ensemble method uses the heuristic described in Section 2.2, and PCA+EM uses BIC. Table 5 reports the average NMI and *k* values for each method averaged over five runs. Because CE is biased toward a larger number of clusters, we only use NMI in this comparison. From Table 5 we see that the cluster ensemble method outperforms PCA+EM for all three data sets. In addition, NMI for the ensemble has lower variance than for PCA+EM. The difference is more significant for HRCT and CHART than EOS, while HRCT and CHART had higher original dimensionality than EOS. From these limited experiments we conjecture that our method is most beneficial when the original dimensionality is large. We want to mention that computationally our method is less efficient than PCA+EM but can be easily parallelized when time is a concern.

4. Analysis of Diversity for Cluster Ensembles

For supervised ensemble approaches, diversity of the base-level classifiers has proven to be a key element in increasing classification performance (Dietterich, 2000). In the relatively new area of unsupervised ensembles, the impact of diversity and quality of the individual clustering solutions on the final ensemble performance has not been fully understood.

To perform this analysis we follow the approach taken by Dietterich (2000) and graph the diversity versus quality for each pair of clustering solutions in the ensemble. To measure diversity, we calculate the NMI *between each pair of clustering solutions.* To obtain a single quality measure for each pair, we average their NMI values as computed between each of the two solutions and the class labels from the labeled data set. Figure 4 shows the diversity-quality diagram for each of the three data sets. Note that when the NMI *between* two solutions (shown on the y axis) is zero the diversity is maximized. In contrast, maximizing the *average* NMI of each pair (shown on the x axis), maximizes their quality. Therefore we want our points to be close to the right-hand bottom corner of each graph. Thirty runs of RP+EM are used to form each ensemble, and five ensembles are formed for each data set. In each graph we also show the average NMI for the five final ensemble solutions as reported in Table 5.

Each of the three data sets shows somewhat different behavior. The left graph shows that the individual clustering solutions for the HRCT data set are highly diverse but have fairly low quality. For the CHART data set (middle), we see that RP+EM formed a set of clustering solutions with a wide range of quality and diversity. In contrast, EOS (right) has slightly higher quality than HRCT but much lower diversity.

In comparing the diversity/quality results to the performance of the entire ensemble (indicated by the dotted line in each graph), we see evidence that for an

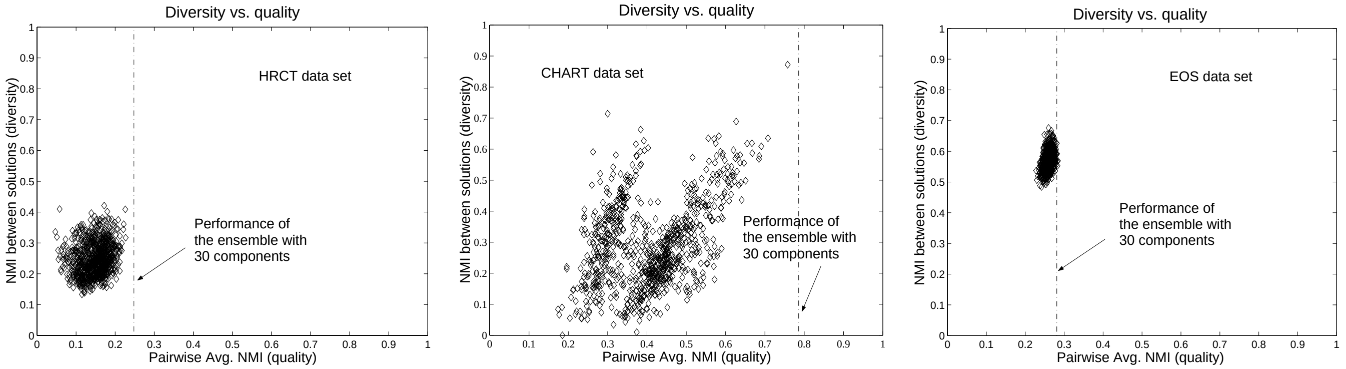

Figure 4. The diversity-quality diagram for three data sets.

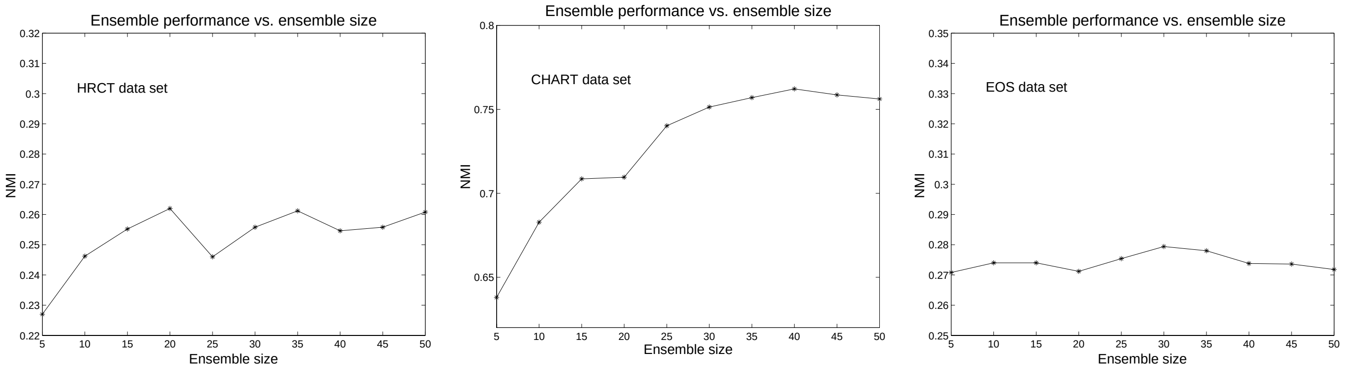

Figure 5. The performance of the ensemble method with different ensemble sizes.

ensemble of size thirty, high diversity leads to greater improvements in the ensemble quality. Specifically, we see the least improvement of the ensemble over a single run of RP+EM for the EOS data set, which has significantly lower diversity than the other two. On the other hand, less improvement is obtained for the HRCT data set in comparison with the CHART data set, which suggests that the quality of individual clustering solutions also limits the performance of a fixed-size ensemble. To gain further insight into these issues, we examined the impact of the ensemble size on performance.

Figure 5 plots the NMI value of the ensemble's final clustering solution for ensemble sizes ranging from five to fifty. The points in the graph are generated as follows. To remove the effect of k on NMI, for each ensemble size r we force our algorithm to produce a fixed number, k, clusters for different values of k as shown in Table 4. The NMI values for different values of k are then averaged to obtain a single performance measure for each ensemble size. We repeat the above process five times and average the results to obtain a stable estimate for the performance measure for each r.

From Figure 5, we can see that increasing the ensemble size helps only for data sets with high diversity. For the CHART data set, We can see a clear and stable trend of performance improvement as the ensemble size increases. For the HRCT data set, we observe a similar but less stable trend. For the EOS data set, the performance gain is negligible as ensemble size increases.

These results suggest that the ensemble performance is strongly influenced by both the quality and the diversity of the individual clustering solutions. If the individual clustering solutions have little diversity, then not much leverage can be obtained by combining them. The quality of the individual solutions limits the performance of a fixed-size ensemble and low quality solutions may cause the ensemble performance to oscillate as the ensemble size changes.

As shown from the experimental results, random projection successfully introduced high diversity into the clustering solutions for both HRCT and CHART data set. This suggests that random projection can produce diverse clustering solutions when the original dimension is high and the features are not highly redundant.[6] An open question is how to improve the quality of the individual clustering solutions. Our future work will

[6] If the features are highly redundant then many random projections will lead to the same clustering.

investigate a tentative solution – evaluate each clustering solution using criteria such as the log-likelihood of the Gaussian model and select only the "good" ones to form the ensemble.

5. Conclusional Remarks

Techniques have been investigated to produce and combine multiple clusterings in order to achieve an improved final clustering. Such methods are formally defined as cluster ensemble methods by Strehl and Ghosh (2002). Due to space limits, please refer to (Strehl & Ghosh, 2002) for a comprehensive survey of the related work. While our work can be considered as an extended instantiation of the general cluster ensemble framework, there are significant distinctions between our work and previous studies in how we form the original clusters, how the clusters are combined and the core problem we are trying to solve – clustering high dimensional data. We conclude here with the major contributions of this work: 1) we examined random projection for high dimensional data clustering and identified its instability problem, 2) we formed a novel cluster ensemble framework based on random projection and demonstrated its effectiveness for high dimensional data clustering, and 3) we identified the importance of the quality and diversity of individual clustering solutions and illustrated their influence on the ensemble performance with empirical results.

Acknowledgment

The authors were supported by NASA under Award number NCC2-1245.

References

Achlioptas, D. (2001). Database-friendly random projections. *Proceedings of the Twentieth ACM Symposium on Principles of Database Systems* (pp. 274–281). ACM Press.

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data* (pp. 94–105). ACM Press.

Bingham, E., & Mannila, H. (2001). Random projection in dimensionality reduction: Applications to image and text data. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 245–250). ACM Press.

Chakrabarti, K., Keogh, E., Mehrotra, S., & Pazzani, M. (2002). Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Transactions on Database Systems*, *27*, 188–228.

Dasgupta, S. (2000). Experiments with random projection. *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)* (pp. 143–151). Morgan Kaufmann.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine learning*, *2*, 139–157.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* John Wiley & Sons.

Dy, J. G., & Brodley, C. E. (2000). Feature subset selection and order identification for unsupervised learning. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 247–254). Morgan Kaufmann.

Dy, J. G., Brodley, C. E., Kak, A., Shyu, C., & Broderick, L. S. (1999). The customized-queries approach to CBIR using EM. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 400–406). IEEE Computer Society Press.

Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, *41*, 578–588.

Fukunaga, K. (1990). *Statistical pattern recognition (second edition).* Academic Press.

Hettich, S., & Bay, S. D. (1999). The UCI KDD archive.

Kaski, S. (1998). Dimensionality reduction by random mapping. *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks* (pp. 413–418). IEEE Neural Networks Council.

Papadimitriou, C. H., Raghavan, P., Tamaki, H., & Vempala, S. (1998). Latent semantic indexing: A probabilistic analysis. *Proceedings of the Seventeenth ACM Symposium on the Principles of Database Systems* (pp. 159–168). ACM press.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*, 846–850.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Machine Learning Research*, *3*, 583–417.