JMLR: Workshop and Conference Proceedings 27:111–122, 2012 Workshop on Unsupervised and Transfer Learning

Stochastic Unsupervised Learning on Unlabeled Data

Chuanren Liu CHUANREN.LIU@RUTGERS.EDU
Rutgers, the State University of New Jersey, Newark, NJ 07102, USA

Jianjun Xie JIANJUNXIE@GMAIL.COM
CoreLogic, 12395 First American Way, Poway, CA 92064, USA

Yong Ge YONGGE@RUTGERS.EDU
Hui Xiong HXIONG@RUTGERS.EDU
Rutgers, the State University of New Jersey, Newark, NJ 07102, USA

Editor: I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver

Abstract

In this paper, we introduce a stochastic unsupervised learning method that was used in the 2011 Unsupervised and Transfer Learning (UTL) challenge. This method is developed to preprocess the data that will be used in the subsequent classification problems. Specifically, it performs K-means clustering on principal components instead of raw data to remove the impact of noisy/irrelevant/less-relevant features and improve the robustness of the results. To alleviate the overfitting problem, we also utilize a stochastic process to combine multiple clustering assignments on each data point. Finally, promising results were observed on all the test data sets. Indeed, this proposed method won us the second place in the overall performance of the challenge.

Keywords: Stochastic Unsupervised Learning, Clustering, K-means, Principal Component Analysis (PCA)

1. Introduction

Data preprocessing is usually critical for the success of building classification models. There are many unsupervised learning techniques which can be exploited for data preprocessing in a complementary way. First, clustering techniques target on dividing data objects into different groups such that the objects in the same cluster are more similar to one another than to those from different clusters. Clustering techniques are widely used for summarizing data objects and capturing key data characteristics (Jain and Dubes, 1988). Among various clustering algorithms, K-means clustering has been identified as one of the top 10 algorithms in data mining by the IEEE International Conference on Data Mining (ICDM) in December 2006 (Wu et al., 2008).

Also, principal Component Analysis (PCA) (Jolliffe, 2002) is an effective technique for dimension reduction and feature preprocessing. It transforms the data into a new coordinate system such that the greatest variance is achieved by projecting the data into the first coordinate (called the first principal component), the second greatest variance achieved into the second coordinate, and so on. Many researchers combined the K-means and PCA together to achieve more stable results (Ben-Hur and Guyon, 2003). It has been shown that

the principal components are the continuous solutions to the discrete cluster membership indicators for K-means clustering (Ding and He, 2004).

The 2011 Unsupervised and Transfer Learning Challenge (Guyon et al., 2011) provided a platform for participants to learn good data representations through data preprocessing that can be re-used across tasks by building models that capture regularities of the input space. The representations are evaluated by the organizers on supervised learning target tasks which are unknown to the participants. In the first phase of the challenge, the competitors are given only unlabeled data to learn their data representation. In the second phase of the challenge, the competitors have available, in addition to unlabeled data, a limited amount of labeled data from source tasks distinct from the target tasks.

In this paper, we present the method that we used in the unsupervised learning challenge (first phase). By exploiting the advantages of both PCA and cluster ensemble techniques, we propose a stochastic unsupervised learning method for data processing. This unsupervised learning method is developed to preprocess the data that will be used in the subsequent binary classification problems. There are two challenging issues for the proposed task. First, there is no labeled data in support of this data preprocessing. Without ground truth, it is difficult to identify noisy or irrelevant features. Second, unsupervised learning methods like K-means start by randomly choosing initial cluster seeds. The results obtained in this way are not only dependent on the chosen seeds, but can also be locally optimal. For the first issue, we use K-means to cluster data represented only with the first P principal components by PCA. In this way, it is expected to remove the negative impact of noisy/irrelevant/less-relevant features. For the second issue, we apply a stochastic strategy to combine clustering results of multiple runs of K-means with random initialization. An ensemble of cluster labels is produced for each data point, which is expected to help alleviate the problem of robustness, clustering quality, and overfitting. This stochastic clustering process has been explored in the semi-supervised learning problems (Xie and Xiong, 2011).

The effectiveness of the data representation obtained by unsupervised learning is evaluated by the organizers on supervised learning tasks (i.e. using labeled data not available to the participants) using Hebbian classifier. Specifically, with the training data matrix $\mathbf{X}$ (one row per instance), the classifier computes the wieght $\boldsymbol{w}$ as $\mathbf{X}^T\boldsymbol{y}$, where $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^T$, $y_i = 1/n_p$ if the ith training instance is positive, $y_i = -1/n_n$ otherwise, where n_p, n_n are the number of positive and negative training examples respectively. The test instance $\boldsymbol{x}$ (column vector) will be classified according to the linear discriminant $\boldsymbol{w}^T\boldsymbol{x}$. It is noted that the size of training data $\mathbf{X}$ is very small (no more than 64 per classification problem) in this challenge. The model performance is reported with the metric of Area under the Learning Curve (ALC) which is referred to as the global score. The participants are ranked by ALC for each individual data set. The winner is determined by the best average rank over all data sets for the results of their last complete experiment. We will see the proposed method is effective especially in such a small training set scenario.

Such a linear discriminant classifier assumes the instances lying in the feature space are linearly separable. However, it is not necessarily true in many real-world data sets. For example, Figure 1 illustrates a situation of a mixture model, where the positive instances indicated by the plus marks are surrounded by 3 groups of negative instances indicated by the circles. Noises are indicated by green dots. With clustering algorithms, we can cluster

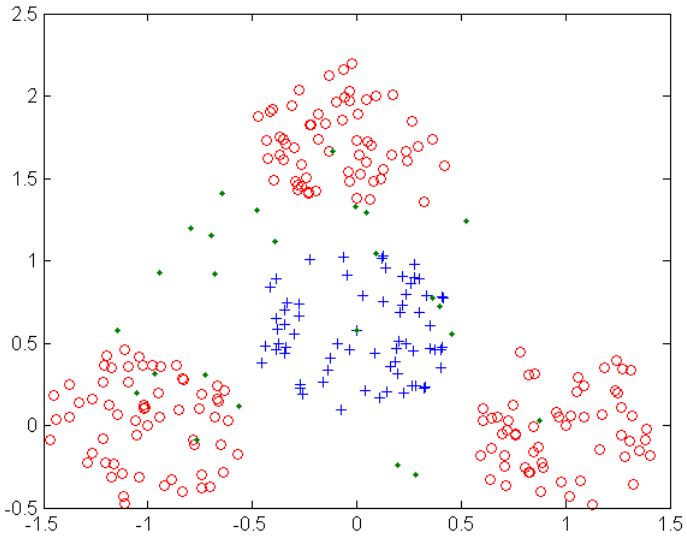


Figure 1: An example of a mixture model.

the data set into 4 groups, whose representation becomes linearly separable by the Hebbian classifier. The above can be reflected in the experimental results.

Overview. The remainder of this paper is organized as follows. In Section 2, we describe the stochastic unsupervised learning method based on K-means and PCA. Section 3 shows the results. In Section 4, we discuss the limitations of the proposed approaches and describe the potential directions for future work.

2. The Stochastic Clustering Algorithm

2.1. The Algorithm

Algorithm 1 details the common strategy we used for all 5 data sets in the challenge. The output is the final data representation, which is a binary representation of derived cluster labels. If K is 3 for a given data set, the binary representations of label 1, 2 and 3 are (1 0 0), (0 1 0) and (0 0 1), respectively. Therefore, our final data representation will be a bagged $N \times KT$ matrix, where N is the number of examples, K is the number of clusters and T is the number of stochastic iterations. Each data element in the matrix is either 1 or 0. Such a binary representation is chosen to eliminate the numeric meaning of clustering labels which is misleading for the Hebbian classifier.

It is noted that the data set $\mathbf{X}$ is not the raw data set. It is the first P principal components of the raw data set. In the challenge, we used different P and necessary variants of naive PCA for each data set based on online feedback from the validation set. For example, instead of analyzing the principal components of covariance matrix for all features, we also tried to decomposing the correlation matrix, which implies dividing by standard deviation prior to computing the covariances. The transformation of the raw data to re-

Algorithm 1: The Stochastic Clustering Algorithm

Input Data set $\mathbf{X}$, the number of clusters K, the number of stochastic iterations T

Output Data set $\mathbf{Y}$

1. For $t = 1, 2, \cdots, T$
   (a) Randomly choose K seeds from $\mathbf{X}$ for K-means to generate clusters. Denote the clustering assignments by $\boldsymbol{I}$.
   (b) Transform $\boldsymbol{I}$ to binary format, i.e. for each assignment

$$i \mapsto \boldsymbol{e_i} = (e_{i1}, e_{i2}, \cdots, e_{iK})$$

   where $e_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$. Denote the binary matrix by

$$\mathbf{B_t} = \begin{pmatrix} \boldsymbol{e_{I_1}} \\ \boldsymbol{e_{I_2}} \\ \vdots \\ \boldsymbol{e_{I_N}} \end{pmatrix}$$

   where I_n is the assignment of the nth instance.
2. Combine $\mathbf{B_t}, t = 1, 2, \cdots, T$ together as $\mathbf{Y} = (\mathbf{B_1}|\mathbf{B_2}|\cdots|\mathbf{B_T})$.

trieved principal components also can be followed by additional processing strategies, such as standardization (to subtract mean and divide deviation for each feature) and weighting (to weight each component by its corresponding eigenvalue).

For the clustering algorithm K-means, the number of clusters was also determined based on online feedback during this challenge. For nearly every data set, we found the real number of classes to be predicted. The only exception is SYLVESTER, where the real numbers of classes in the validation set and the final set are 2 and 3 respectively, and we used 3 as the number of clusters. In addition to K, we also used different distance/similarity metrics for different data sets. Basically, for low dimensional $\mathbf{X}$, Euclidean distance is used. Otherwise in the high dimensional case, cosine similarity is preferred. Cosine performs an implicit instance-level standardization, i.e., the instance vector is normalized to be of unit length. We found feature-level standardization could also improve the clustering results, such as HARRY. For TERRY, which is in text recognition domain, the well-known TF-IDF transformation is used prior to computing cosine similarities.

Another parameter in Algorithm 1 is the number of stochastic iterations. The motivation of the stochastic process is to settle the overfitting phenomenon. Although the binary matrix generated from only one clustering solution can be directly fed to the classifier, the final classification result will vary a lot with the clustering solution. By combining several different clustering solutions, we found the final classification result could be improved better

than that based on any single clustering solution. Ideally, the final result will converge along with the increasing number of stochastic iterations. When the result becomes stable on the validation set, we believe the overfitting problem on the final set has also been circumvented. We analyzed the results of $20, 40, 60, 80$ and 100 iterations. For most of the data sets, stable results are observed on the validation set after 60 iterations. Thus, in our submission to the challenge, we set the number of stochastic iterations as 100. Details of these variations on each data set will be described in Section 3.

2.2. Cluster Assumption

In fact, the proposed algorithm maps the data from the original data space to the space discovered by the underlying clustering algorithm, in the hope that the class does not change in regions of high density within clusters. Such a cluster assumption can be explained using cluster kernels. Specifically, with the achieved clustering solution

$$l : \boldsymbol{z} \mapsto l(\boldsymbol{z}) \in \{1, 2, \cdots, K\}$$

where $l(\boldsymbol{z})$ is the clustering assignment for any clustered instance $\boldsymbol{z}$, the mapping function is

$$\phi(\boldsymbol{z}) = ([l(\boldsymbol{z}) = 1], [l(\boldsymbol{z}) = 2], \cdots, [l(\boldsymbol{z}) = K])^T,$$

and the inner product kernel

$$\phi(\boldsymbol{x})^T \phi(\boldsymbol{z}) = [l(\boldsymbol{x}) = l(\boldsymbol{z})]$$

will be used by Hebbian classifier. By combining multiple clustering solutions together, the inner product of $\boldsymbol{x}$ and $\boldsymbol{z}$ in the mapped space is $\sum_{t=1}^{T} [l_t(\boldsymbol{x}) = l_t(\boldsymbol{z})]$, where l_t is the tth clustering solution. Such a combination is also used in the study of consensus clustering (Hu and Sung, 2005).

2.3. An Illustrative Example

To illustrate the effectiveness and rationale of the proposed algorithm, especially of the clustering component, we analyzed the results of 100 runs of K-means for the toy data set ULE whose true labels are available. For each clustering solution, in addition to mean squared error (MSE) as the clustering criterion function, we also computed ALC and purity (the fraction of correctly classified data when all data in each cluster is classified as the majority class in that cluster). As shown by the representative solutions in Table 1, one can see that better classification results really come along with better clustering solutions. The best ALC value of 0.83764 is achieved with the lowest MSE value of 0.9102239 and the highest purity value of 0.93872. More interestingly, by combining all 100 clustering solutions, we can achieve an ALC value of 0.86642, which is significantly better than that of the best single solution. Such an ensemble effect is the key motivation of the proposed algorithm.

3. Results

In this section, we provide an empirical study of the proposed stochastic unsupervised learning method. In most of the data sets studied, the proposed method achieves better performances than that of the raw data and PCA.

Table 1: A Comparison of MSE, purity and ALC.

MSE	**Purity**	**ALC**
0.9102239	0.93872	0.83764
0.9782505	0.65332	0.51611
0.9959307	0.64429	0.51231
1.0049960	0.63550	0.48539
1.0049971	0.63550	0.48545
1.0050026	0.63501	0.48563
1.0050027	0.63452	0.48586
1.0050034	0.63599	0.48262

Table 2: The Results on AVICENNA.

	Validation	**Final**	**Algorithm details**
Raw Data	0.1034	0.1501	Original data
PCA	0.1386	0.1906	First 50 standardized PCs from Covariance Matrix and First 50 standardized PCs from Correlation Matrix
K-Means	0.1668	0.1511	Stochastic K-means on first 100 standardized PCs. Cluster number = 5.

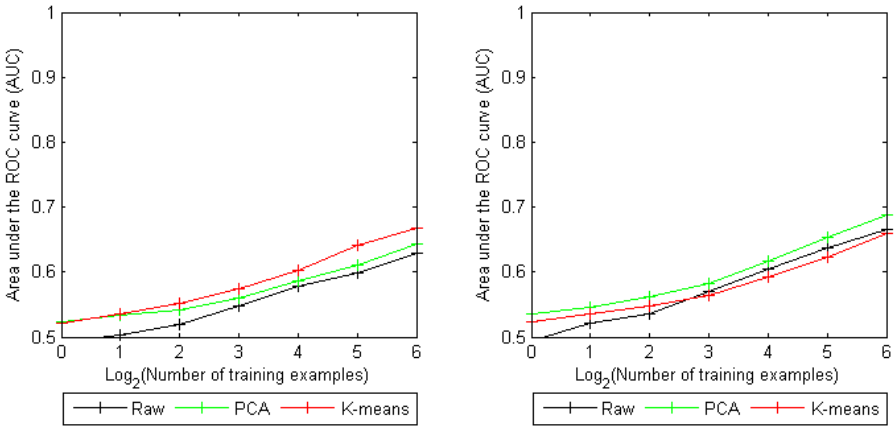
3.1. AVICENNA: Arabic manuscripts

The results on the AVICENNA data set are shown in Table 2. It seems difficult to get good results on either the validation set or the final set, for the best global score on the leader board turns out less than 0.2 for the validation set. This is the only data set in our experiments where the PCA itself has better global scores on the final data set than K-means. We believe this is due to the label overlaps in this data set; that is, one example can belong to multiple classes.

The learning curves of the three scenarios (raw data, PCA and K-Means) are shown in Figure 2. The PCs are standardized for this data set such that each feature has zero mean and unit variance. We did notice that the K-means underperforms PCA during the first phase of the challenge through the on-line feedback on the validation set. Therefore, we chose the PCA results as our final experiment. However, we did some improvements on K-means during the second phase. We found that if we first did a record level normalization on each variables (this is equivalent to a Term-Frequency transformation in document classification), then did PCA on the normalized variables and stochastic K-means on the first 100 PCs, we could lift the global score on the validation set from 0.1386 to 0.1668. Unfortunately, this improvement on the validation set did not hold on the final set. Our K-means results on the final set actually dropped to 0.1511 from 0.1906.

3.2. HARRY: Human action recognition

Table 3 lists our experimental results on both PCA and K-means on the HARRY data set. In the table, we can observe that PCA works very well on the validation set. The first 5 weighted PCs (weighted by the corresponding eigenvalue of each principal component) can achieve a 0.8056 global score in the validation set. For this data, which is high dimensional

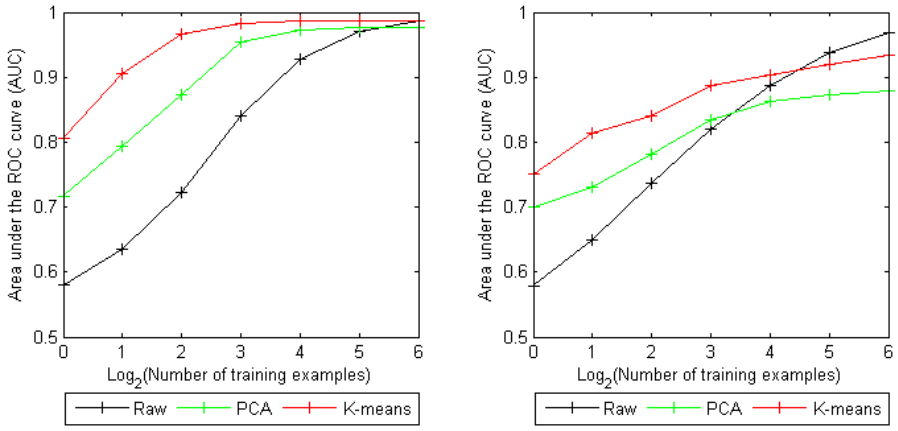


(*a*) Validation (*b*) Final

Figure 2: The learning curve on AVICENNA

Table 3: The Results on HARRY.

	Validation	**Final**	**Algorithm details**
Raw Data	0.6264	0.6017	Original data
PCA	0.8056	0.6243	First 5 weighted PCs from correlation matrix
K-Means	0.9085	0.7357	Stochastic K-means on standardized data. Cluster number = 3.



(*a*) Validation (*b*) Final

Figure 3: The Learning curves on HARRY

and very sparse, the stochastic K-means on standardized data works better than that on PCs. We used the Cosine similarity as the distance measure in K-means clustering. The number of clusters is set to 3. In Table 3, we can see that, while PCA works pretty well, the stochastic K-means without PCA works much better. Such a phenomenon was also observed in TERRY, which is also high dimensional and very sparse.

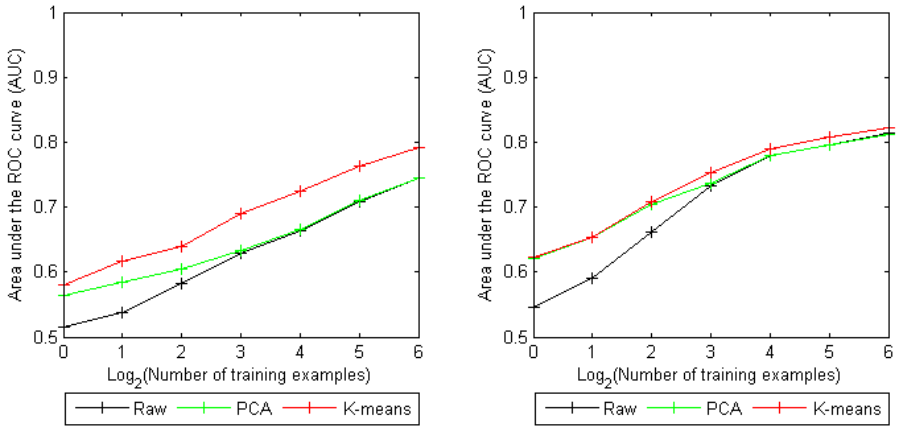
The learning curves on both the validation and the final sets are illustrated in Figure 3. We can see that the improvements on the validation set do hold well on the final set.

3.3. RITA: Object recognition

Table 4: The results on RITA.

	Validation	**Final**	**Algorithm details**
Raw Data	0.2504	0.4133	Original data
PCA	0.2834	0.4622	First 50 PCs from covariance matrix
K-Means	0.3737	0.4782	Stochastic K-means on standardized 50 PCs. Cluster number = 3.

RITA is another difficult data set in addition to AVICENNA. Our experimental results on both PCA and K-means on the RITA data set are shown in Table 4. The first 50 principal components achieve ALC = 0.2834, 0.4622 on the validation set and the final set, respectively. The stochastic K-means gives the best results. We find that Euclidian distance is better than the Cosine similarity in this case. The number of clusters is set to 3. The learning curves on both validation and final sets are illustrated in Figure 4. We can see that the improvements on the global score from PCA and K-means over the raw data mainly come from the beginning of the learning curve, which may correspond to small number of training samples.



(*a*) Validation (*b*) Final

Figure 4: The learning curve on RITA

Table 5: The results on SYLVESTER.

	Validation	**Final**	**Algorithm details**
Raw Data	0.2167	0.3095	Original data
PCA	0.5873	0.4436	First 7 standardized PCs from correlation matrix
K-Means	0.7146	0.5828	Stochastic K-means on standardized 15 PCs. Cluster number = 3.

3.4. SYLVESTER: Ecology

Table 5 lists our experimental results on both PCA and K-means on the SYLVESTER data set. SYLVESTER has only 100 features and is not sparse. In the table, we can see that the first 7 principal components can do much better than the original data. The stochastic K-means using $K = 3$ further improves the PCA results from 0.4436 to 0.5828 on the final set. Indeed, our result on the final set was ranked No. 1 in the first phase of the challenge.

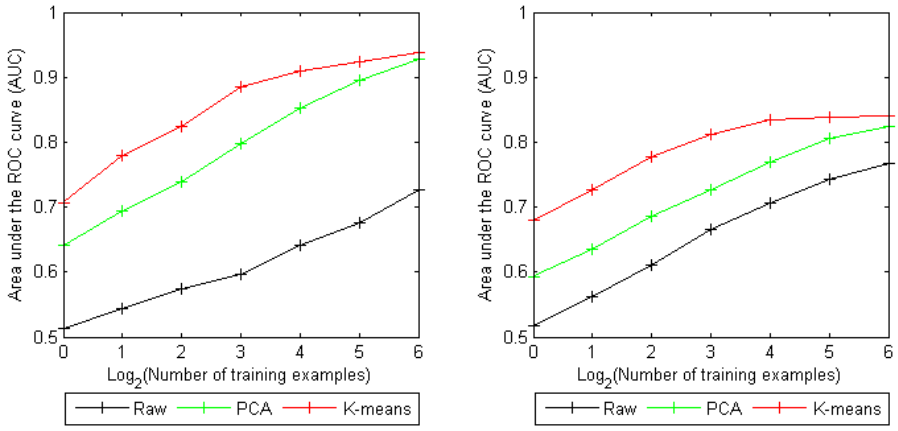
Also, Figure 5 shows the learning curves on both the validation set and the final set. In the figure, a similar trend of performances can be observed as in Table 5.

3.5. TERRY: Text recognition

Table 6: The results on TERRY.

	Validation	**Final**	**Algorithm details**
Raw Data	0.6969	0.7550	Original data
PCA	0.7949	0.8317	First 5 PCs from covariance matrix
K-Means	0.8176	0.8437	Stochastic K-means on TF-IDF data. Cluster number = 5.

Table 6 lists our experimental results on both PCA and K-means on the TERRY data set. This is another high dimensional and very sparse data set similar to HARRY. We find



(*a*) Validation (*b*) Final

Figure 5: The learning curve on SYLVESTER.

Table 7: A Comparison of our results with the overall winner's results. The winner is determined by the average rank on all 5 final data sets. Our results are ranked No. 2.

Data	**Winner-Valid**	**Winner-Final**	**Winner-Rank**	**Our-Valid**	**Our-Final**	**Our-Rank**
AVICENNA	0.1744	0.2183	1	0.1386	0.1906	6
HARRY	0.8640	0.7043	6	0.9085	0.7357	3
RITA	0.3095	0.4951	1	0.3737	0.4782	5
SYLVESTER	0.6409	0.4569	6	0.7146	0.5828	1
TERRY	0.8195	0.8465	1	0.8176	0.8437	2

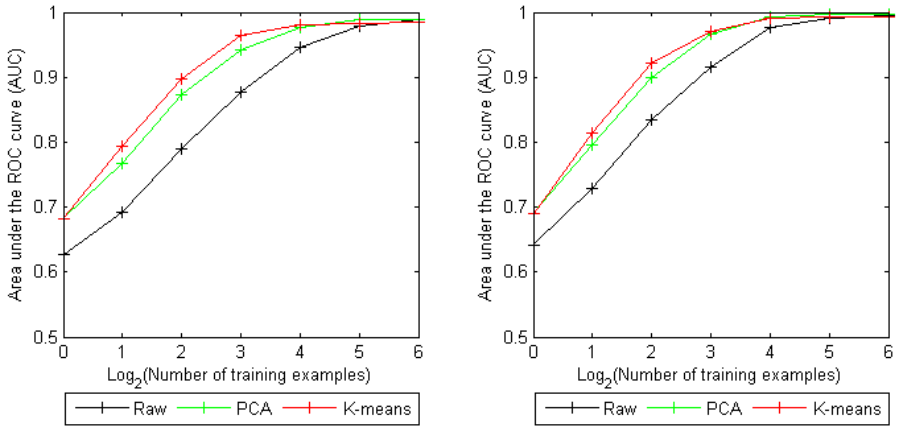
the PCA can generate much better results than the original data like those of HARRY. The first 5 principal components can achieve a global score of 0.8317 on the final set.

However, standardization does not help the clustering anymore. The TERRY data set is from the text recognition domain, where the TF-IDF weight (term frequency-inverse document frequency) is often used for information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. Thus, instead of standardization, we first did TF-IDF transformation on the raw data, then did stochastic K-means using $K = 5$ based on the Cosine similarity.

The learning curves on the validation and final sets are illustrated in Figure 6. We can see that the improvements on the validation set hold well on the final set. The greatest lift on the learning curve over raw data happens in the middle range of the x-axis. Our results on the final set is ranked No. 2 in the challenge.

4. Discussion

In this section, we first show a comparison of the proposed method with the overall winner. Then, we conclude this study by discussing its limitations and potential extensions.



(*a*) Validation (*b*) Final

Figure 6: The learning curve on TERRY.

First, by combining PCA and K-means, the proposed stochastic unsupervised learning method achieves the stable results on most of the data sets. Table 7 lists our global scores on all 5 data sets against those of the overall winner. We are ranked 1st on SYLVESTER, 2nd on TERRY and 3rd on HARRY. Although the results on AVICENNA and RITA are not impressive in rank compared to others, they are within about 0.02 in ALC from the winner's results. Our overall performance in rank was placed 2nd in the challenge.

Indeed, the performance of the proposed method significantly depends on its component: K-means clustering. Although we employed a stochastic strategy of cluster ensemble, the inherent characteristics of K-means still have impact on the final results. For instance, since K-means tends to favor globular clusters with similar sizes (Xiong et al., 2006, 2009), it cannot handle some of the data sets in the challenge that have different shapes or sizes of clusters. Also, K-means is very sensitive to data density. In the case that data have various densities, some density based clustering algorithms, such as DBSCAN (Ester et al., 1996), could be used in the proposed method. Moreover, some fuzzy clustering methods could be used to handle the data with overlapping labels (Nock and Nielsen, 2006), such as AVICENNA. Finally, when the labels of the data sets are available, we can explore the relationship between the quality of the clustering results and the accuracy of the final classification results. Such information may help to make informed decision in both generation and combination phases of cluster ensemble.

Acknowledgments

First, we would like to thank the challenge organizers for setting up an excellent platform and providing real-world data for this challenge. This research was supported in part by National Science Foundation (NSF) via grant number CCF-1018151.

References

A. Ben-Hur and I. Guyon. Detecting stable clusters using principal component analysis. *Methods in Molecular Biology*, 224:159–182, 2003.

C. Ding and X. He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.

M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. Portland: AAAI Press, 1996.

I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. W. Aha. Unsupervised and transfer learning challenge. In *Proc. IJCNN*, 2011.

T. Hu and S.Y. Sung. Consensus clustering. *Intelligent Data Analysis*, 9(6):551–565, 2005.

A.K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

I. Jolliffe. Principal component analysis. *Encyclopedia of Statistics in Behavioral Science*, 2002.

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

R. Nock and F. Nielsen. On weighting clustering. *IEEE transactions on pattern analysis and machine intelligence*, pages 1223–1235, 2006.

X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

Jianjun Xie and Tao Xiong. Stochastic semi-supervised learning on partially labeled imbalanced data. In *Proc. AISTATS Workshop on Active Learning and Experimental Design*, pages 85–98, 2011.

H. Xiong, J. Wu, and J. Chen. K-means clustering versus validation measures: a data distribution perspective. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 779–784. ACM, 2006.

H. Xiong, J. Wu, and J. Chen. K-means clustering versus validation measures: a data-distribution perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):318–331, 2009.