Combining Multiple Clustering Systems

Constantinos Boulis and Mari Ostendorf

Department of Electrical Engineering,
University of Washington, Seattle, WA 98195, USA
boulis,mo@ee.washington.edu

Abstract. Three methods for combining multiple clustering systems are presented and evaluated, focusing on the problem of finding the correspondence between clusters of different systems. In this work, the clusters of individual systems are represented in a common space and their correspondence estimated by either "clustering clusters" or with Singular Value Decomposition. The approaches are evaluated for the task of topic discovery on three major corpora and eight different clustering algorithms and it is shown experimentally that combination schemes almost always offer gains compared to single systems, but gains from using a combination scheme depend on the underlying clustering systems.

1 Introduction

Clustering has an important role in a number of diverse fields, such as genomics [1], lexical semantics [2], information retrieval [3] and automatic speech recognition [4], to name a few. A number of different clustering approaches have been suggested [5] such as agglomerative clustering, mixture densities and graph partitioning. Most clustering methods focus on individual criteria or models and do not address issues of combining multiple different systems. The problem of combining multiple clustering systems is analogous to the classifier combination problem, that has received increased attention over the last years [6]. Unlike the classifier combination problem, though, the correspondence between clusters of different systems is unknown. For example, consider two clustering systems applied to nine data points and clustered in three groups. System A's output is $\boldsymbol{o}_A = [1, 1, 2, 3, 2, 2, 1, 3, 3]$ and system B's output is $\boldsymbol{o}_B = [2, 2, 3, 1, 1, 3, 2, 1, 1]$, where the i-th element of $\boldsymbol{o}$ is the group to which data point i is assigned. Although the two systems appear to be making different decisions, they are in fact very similar. Cluster 1 of system A and cluster 2 of system B are identical, and cluster 2 of system A and cluster 3 of system B agree 2 out of 3 times, as cluster 3 of system A and cluster 1 of system B. If the correspondence problem is solved then a number of system combination schemes can be applied.

Finding the optimum correspondence requires a criterion and a method for optimization. The criterion used here is maximum agreement, i.e. find the correspondence where clusters of different systems make the maximum number of the same decisions. Second, we must optimize the selected criterion. Even if we assume a 0 or 1 correspondence between clusters with only two systems of M

topics each, a brute-force approach would require the evaluation of $M!$ possible solutions. In this work, three novel methods are presented for determining the correspondence of clusters and combining them. Two of the three methods are formulated and solved with linear optimization and the third uses singular value decomposition.

Another contribution of this work is the empirical result that the combination schemes are not independent of the underlying clustering systems. Most of the past work has focused on combining systems generated from a single clustering algorithm (using resampling or different initial conditions), usually k-means. In this work, we experimentally show that the relative gains of applying a combination scheme are not the same across eight different clustering algorithms. For example, although the mixture of multinomials was one of the worse performing clustering algorithms, it is shown that when different runs were combined it achieved the best performance of all eight clustering algorithms in two out of three corpora. The results suggest that an algorithm should not be evaluated solely on the basis of its individual performance, but also on the combination of multiple runs.

2 Related Work

Combining multiple clustering systems has recently attracted the interest of several researchers in the machine learning community. In [7], three different approaches for combining clusters based on graph-partitioning are proposed and evaluated. The first approach avoids the correspondence problem by defining a pairwise similarity matrix between data points. Each system is represented by a $D \times D$ matrix (D is the total number of observations) where the (i, j) position is either 1 if observations i and j belong to the same cluster and 0 otherwise. The average of all matrices is used as the input to a final similarity-based clustering algorithm. The core of this idea also appears in [8–12]. A disadvantage of this approach is that it has quadratic memory and computational requirements. Even by exploiting the fact that each of the $D \times D$ matrices is symmetric and sparse, this approach is impractical for high D.

The second approach taken in [7], is that of a hypergraph cutting problem. Each one of the clusters of each system is assumed to be a hyperedge in a hypergraph. The problem of finding consensus among systems is formulated as partitioning a hypergraph by cutting a minimum number of hyperedges. This approach is linear with the number of data points, but requires fairly balanced data sets and all hyperedges having the same weight. A similar approach is presented in [13], where each data point is represented with a set of meta-features. Each meta-feature is the cluster membership for each system, and the data points are clustered using a mixture model. An advantage of [13] is that it can handle missing meta-features, i.e. a system failing to cluster some data points. Algorithms of this type, avoid the cluster correspondence problem by clustering directly the data points.

The third approach presented in [7], is to deal with the cluster correspondence problem directly. As stated in [7], the objective is to *"cluster clusters"*, where each cluster of a system is a hyperedge and the objective is to combine similar hyperedges. The data points will be assigned to the combined hyperedge they most strongly belong to. Clustering hyperedges is performed by using graph-partitioning algorithms. The same core idea can also be found in [10, 14–16]. In [10], different clustering solutions are obtained by resampling and are aligned with the clusters estimated on all the data. In both [14, 15], the different clustering solutions are obtained by multiple runs of the k-means algorithm with different initial conditions. An agglomerative pairwise cluster merging scheme is used, with a heuristic to determine the corresponding clusters. In [16], a two-stage clustering procedure is proposed. Resampling is used to obtain multiple solutions of k-means. The output centroids from multiple runs are clustered with a new k-means run. A disadvantage of [16] is that it requires access to the original features of the data points, while all other schemes do not. Our work falls in the third approach, i.e. attempts to first find a correspondence between clusters and then combine clusters without requiring the original observations.

3 Finding Cluster Correspondence

In this paper, three novel methods to address the cluster correspondence problem are presented. The first two cast the correspondence problem as an optimization problem, and the third method is based on singular value decomposition.

3.1 Constrained and Unconstrained Search

We want to find the assignment of clusters to entities (metaclusters) such that the overall agreement among clusters is maximized. Suppose $\boldsymbol{R}_{\{c,s\}}$ is the $D \times 1$ vector representation of cluster c of system s (with D being the total number of documents). The k-th element of $\boldsymbol{R}_{\{c,s\}}$ is $p(cluster = c|observation = k, system = s)$. The agreement between clusters $\{c, s\}$ and $\{c', s'\}$ is defined as:

$$g_{\{c,s\},\{c',s'\}} = \boldsymbol{R}_{\{c,s\}}^T \cdot \boldsymbol{R}_{\{c',s'\}} \tag{1}$$

In addition, suppose that $\lambda_{\{c,s\}}^{\{m\}} = 1$ if cluster c of system s is assigned to metacluster m and 0 otherwise, and $r_{\{c,s\}}^{\{m\}}$ is the "reward" of assigning cluster c of system s to metacluster m, defined as:

$$r_{\{c,s\}}^{\{m\}} = \frac{1}{|I(m)|} \sum_{\{c',s'\} \in I(m)} g_{\{c,s\},\{c',s'\}} \quad , \{c', s'\} \in I(m) \iff \lambda_{\{c',s'\}}^{\{m\}} \neq 0 \tag{2}$$

We seek to find the argument that maximizes:

$$\boldsymbol{\lambda}^* = arg \max_{\boldsymbol{\lambda}} \sum_{m=1}^{M} \sum_{s=1}^{S} \sum_{c=1}^{C_s} \lambda_{\{c,s\}}^{\{m\}} r_{\{c,s\}}^{\{m\}} \tag{3}$$

$$\text{subject to the constraints } \sum_{m=1}^{M} \lambda_{\{c,s\}}^{\{m\}} = 1, \quad \forall c, s \tag{4}$$

Optionally, we may want to add the following constraint:

$$\sum_{c=1}^{C_s} \lambda_{\{c,s\}}^{\{m\}} = 1, \quad \forall s, m \tag{5}$$

This is a linear optimization problem and efficient techniques exist for maximizing the objective function. In our implementation, the GNU Linear Programming library was used.[1] The scheme that results from omitting the constraints of equation (5) is referred to as *unconstrained*, while including them results in the *constrained* combination scheme. The added constraints ensure that exactly one cluster from each system is assigned to each metacluster and are useful when $C_s = C \; \forall s$. The entire procedure is iterative, starting from an initial assignment of clusters to metaclusters and alternating between equations (2) and (3).

The output of the clustering procedure is matrix $\mathbf{F}$ of size $D \times M$, where each column is the centroid of each metacluster. The $\boldsymbol{F}_m$ column is given by:

$$\boldsymbol{F}_m = \frac{1}{|I(m)|} \sum_{\{c,s\} \in I(m)} \boldsymbol{R}_{\{c,s\}}^T \tag{6}$$

This can be the final output or a clustering stage can be applied using the $\mathbf{F}$ matrix as the observation representations. Note that the assignments can be continuous numbers between 0 and 1 (soft decisions) and that the systems do not need to have the same number of clusters, nor do the final number of metaclusters need to be the same as the number of clusters. To simplify the experiments, here we have assumed that the number of clusters is known and equal to the number of topics, i.e. $C_s = M = \#\text{of topics } \forall s$. The methodology presented here does not assume access to the original features and therefore it can be applied in cases irrespective of whether the original features were continuous or discrete.

The optimization procedure is very similar to any partition-based clustering procedure trained with the Expectation-Maximization algorithm, like k-means. In fact, this scheme is "clustering clusters", i.e. expressing clusters in a common vector space and grouping them into similar sets. Although the problem is formulated from the optimization perspective, any clustering methodology can be applied (statistical, graph-partitioning). However, there are two reasons that favor the optimization approach. First, it directly links the correspondence problem to an objective function that can be maximized. Second, it allows us to easily integrate constraints during clustering such as equation (5). As it is shown in section 5, the constrained clustering scheme offers gains over the unconstrained case, when it is appropriate for the task.

3.2 Singular Value Decomposition Combination

The third combination approach we introduce is based on Singular Value Decomposition (SVD). As before, we will assume that all systems have the same

[1] http://www.gnu.org/software/glpk/glpk.html

number of clusters for notational simplicity, though it is not required of the algorithm. Just as before, we construct matrix $\mathbf{R}$ of size $D \times SC$ (D is the number of observations, S is the number of systems, C the number of clusters), where each row contains the cluster posteriors of all systems for a given observation. $\mathbf{R}$ can be approximated as $\mathbf{R} \approx \mathbf{U} * \mathbf{S} * \mathbf{\Lambda^t}$ where $\mathbf{U}$ is orthogonal and of size $D \times C$, $\mathbf{S}$ is diagonal and of size $C \times C$ and $\mathbf{\Lambda}$ is orthogonal and of size $(SC) \times C$. The final metaspace is $\mathbf{R} * \mathbf{\Lambda}$ of size $D \times C$. If we define $p_s(c|d) = p(cluster = c|observation = d, system = s),\ c = 1 \dots C, s = 1 \dots S, d = 1 \dots D$ and $h_C(l) = l - C\lfloor l/C \rfloor$ (remainder of division), then the $\phi_{d,c}$ element of $\mathbf{R} * \mathbf{\Lambda}$ is given by:

$$\phi_{d,c} = \sum_{k=1}^{S} \lambda_{g_c(k),c} p_k \left(h_C(g_c(k)) | d \right) \tag{7}$$

where $g_c(\cdot)$ is a function that aligns clusters of different systems and is estimated by SVD. In essence, SVD identifies the most correlated clusters, i.e. finds $g_c(\cdot)$ and combines them with linear interpolation. The λ weights provide a soft alignment of clusters. After SVD, a final clustering is performed using the $\phi_{d,c}$ representation.

4 Evaluating Clustering Systems

There is no consensus in the literature on how to evaluate clustering decisions. In this work, we used two measures to evaluate the clustering output. The first is the classification accuracy of a one-to-one mapping between clusters and true classes. The problem of finding the optimum assignment of M clusters to M classes can be formulated and solved with linear programming. If $r_{i,j}$ is the "reward" of assigning cluster i to class j (which can be the number of observations they agree), $\lambda_{i,j}$=1 if cluster i is assigned to class j and 0 otherwise are the parameters to estimate, then we seek to find: $\max_{\lambda_{i,j}} \sum_{i,j} r_{i,j} \lambda_{i,j}$ under the constraints $\sum_i \lambda_{i,j} = 1$ and $\sum_j \lambda_{i,j} = 1$. The constraints will ensure a one-to-one mapping.

The second measure we used is the normalized mutual information (NMI) between clusters and classes, introduced in [7]. The measure does not assume a fixed cluster-to-class mapping but rather takes the average mutual information between every pair of cluster and class. It is given by:

$$NMI = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} n_{i,j} \log \left(\frac{n_{i,j} D}{n_i m_j} \right)}{\sqrt{\sum_{i=1}^{M} n_i \log \frac{n_i}{D} \sum_{j=1}^{M} m_j \log \frac{m_j}{D}}} \tag{8}$$

where $n_{i,j}$ is the number of observations cluster i and class j agree, n_i is the number of observations assigned to cluster i, m_j the number of observation of class j and D the total number of observations. It can be shown that $0 < NMI \leq 1$ with NMI=1 corresponding to perfect classification accuracy.

5 Experiments

The multiple clustering system combination schemes that are introduced in this paper are general and can, in principle, be applied to any clustering problem. The task we have chosen to evaluate our metaclustering schemes is topic discovery, i.e. clustering documents according to their topic. Topic discovery is an especially hard clustering problem because of the high dimensionality of the data points and the redundancy of many features. To simplify our experiments, the number of topics is assumed to be known. This is an assumption that is not true in many practical cases, but standard techniques such as Bayesian Information Criterion [17] can be used to select the number of topics. It should be noted that the unconstrained and SVD combination schemes do not require the same number of clusters for all systems. On the other hand, the constrained clustering scheme was proposed based on this assumption.

5.1 Corpora

The techniques proposed in this work are applied on three main corpora with different characteristics. The first corpus is 20Newsgroups[2], a collection of 18828 postings into one of 20 categories (newsgroups). The second corpus is a subset of Reuters-21578[3], consisting of 1000 documents equally distributed among 20 topics. The third corpus is Switchboard-I release 2.0 [18], a collection of 2263 5-minute telephone conversations on 67 possible topics. Switchboard-I and to a smaller extent 20Newsgroups, are characterized with a spontaneous, less structured style. On the other hand, Reuters-21578 contains carefully prepared news stories for broadcasting. 20Newsgroups and the subset of Reuters are balanced, i.e. documents are equally divided by topics, but Switchboard-I is not. Also, the median length of a document varies significantly across corpora (155 words for 20Newsgroups, 80 for the subset of Reuters-21578 and 1328 for Switchboard-I). Standard processing was applied in all corpora. Words in the default stoplist of CLUTO (total 427 words) are removed, the remaining stemmed and only tokens with T or more occurrences (T=5 for 20Newsgroups, T=2 for Reuters-21578 and Switchboard-I) are retained. These operations result in 26857 unique tokens and 1.4M total tokens in 20Newsgroups, 4128 unique tokens and 50.5K total tokens in Reuters, and 11550 unique and 0.4M total tokens in Switchboard.

5.2 Clustering Algorithms

A number of different clustering systems were used, including the mixture of multinomials (MixMulti) and the optimization-based clustering algorithms and criteria described in [19]. The MixMulti algorithm clusters documents by estimating a mixture of multinomial distributions. The assumption is that each topic is characterized by a different multinomial distribution, i.e. different counts

[2] http://www.ai.mit.edu/~jrennie/20Newsgroups/
[3] http://www.daviddlewis.com/resources/testcollections/

Table 1. Performance of different combination schemes on various clustering algorithms for 20Newsgroups.

	Single Run	**Best of 100 runs**	**SVD Combin.**	**Constr. Combin.**	**Unconstr. Combin.**	**No Combin.**
$\boldsymbol{I_1}$						
Accuracy	.422	.412	.418	.417	.408	.459
NMI	.486	.485	.481	.480	.463	.500
$\boldsymbol{I_2}$						
Accuracy	.575	.603	.634	.615	.639	.624
NMI	.601	.621	.637	.628	.640	.637
$\boldsymbol{E_1}$						
Accuracy	.579	.604	.648	.641	.610	.635
NMI	.588	.606	.639	.631	.628	.633
$\boldsymbol{G_1}$						
Accuracy	.535	.561	.581	.562	.578	.576
NMI	.561	.585	.593	.581	.582	.589
$\boldsymbol{G_1'}$						
Accuracy	.576	.608	.642	.630	.563	.644
NMI	.584	.603	.631	.622	.620	.632
$\boldsymbol{H_1}$						
Accuracy	.570	.584	.636	.641	.549	.642
NMI	.593	.610	.629	.627	.592	.628
$\boldsymbol{H_2}$						
Accuracy	.586	.611	.656	.639	.602	.641
NMI	.598	.616	.646	.634	.628	.638
MixMulti						
Accuracy	.534	.620	.679	.677	.621	.651
NMI	.587	.625	.662	.656	.651	.662

of each word given a topic. The probability of a document d is given by: $p(d) \propto \sum_{c=1}^{M} p(c) \prod_{w \in W_d} p(w|c)^{n(w,d)}$ where M is the number of topics, W_d is the set of unique words that appear in document d, $p(w|c)$ is the probability of word w given cluster c and $n(w, d)$ is the count of word w in document d. The cluster c that each document is generated from is assumed to be hidden. Training such a model is carried out using the Expectation-Maximization algorithm [20]. In practice, smoothing the multinomial distributions is necessary. The mixture of multinomials algorithm is the unsupervised analogue of the Naive Bayes algorithm and has been successfully used in the past for document clustering [21]. Mixture models, in general, have been extensively used for data mining and pattern discovery [22].

The software package CLUTO[4] was used for the optimization-based algorithms. Using CLUTO, a number of different clustering methods (hierarchical, partitional and graph-partitioning) and criteria can be used. For example, the I_2 criterion maximizes the function $\sum_{k=1}^{M} \sqrt{\sum_{\boldsymbol{u},\boldsymbol{v} \in c_k} \cos(\boldsymbol{u}, \boldsymbol{v})}$, where c_k is the

[4] http://www-users.cs.umn.edu/~karypis/cluto/

set of documents in cluster k and $\boldsymbol{u}, \boldsymbol{v}$ are the tfidf vector representations of documents u, v respectively. The I_2 criterion attempts to maximize intra-cluster similarity. Other criteria, like E_1, attempt to minimize inter-cluster similarity and yet other criteria, like H_2, attempt to optimize a combination of both. For more information on the optimization criteria and methods, see [19].

Having determined the clustering algorithms to use, the next question is how to generate the systems to be combined. We may combine systems from different clustering algorithms, pick a single algorithm and generate different systems by resampling, or pick a single algorithm and use different initial conditions for each system. In this work we chose the last option.

5.3 Results

On all results reported in this work the *direct* clustering method was used for the CLUTO algorithms. For the *single run* case, the number reported is the average of 100 independent runs. For the *best of 100 runs* case, the number is the average of 10 runs where each run selects the system with the highest objective function out of 100 trials. A trial is an execution of a clustering algorithm with a different initial condition. For the metaclustering schemes, the final clustering is performed, with the default values of CLUTO. 100 runs of the CLUTO algorithm are performed and the one with the highest objective function selected.

In Table 1, the performance of the three combination schemes applied on eight different clustering algorithms on 20Newsgroups is shown. For every clustering algorithm except I_1, we can observe significant gains of the combination schemes compared to a single run or selecting the system with the highest objective function. The results show that the SVD combination outperforms the constrained combination which in turn outperforms the unconstrained combination. This suggests that the constraints introduced are meaningful and lead to improved performance over the unconstrained scheme. Also shown in Table 1 are the results from not using any combination scheme. This means that the clusters of different systems are not combined but rather the cluster posteriors for all systems are used as a new document representation. This corresponds to using matrix $\mathbf{R}$ from subsection 3.2 without any dimensionality reduction. This is the approach taken in [13]. From Table 1, we see that for the MixMulti case there are gains from using SVD combination rather than using no combination of clusters at all. For other systems, gains are small or differences are insignificant, except for I_1 again where accuracy degrades significantly.

In Table 2, the performance of the three combination schemes over the same eight algorithms on a 1000-document subset of Reuters-21578 is shown. The same trends as in Table 1 seem to hold. Combination appears to offer significant improvements for all clustering algorithms, with SVD combination having a lead over the other two combination schemes. In most cases, SVD combination is better than the best individual clustering system. As in Table 1, the constrained scheme is superior to unconstrained but not as good as SVD combination.

In Table 3 the experiments are repeated for the Switchboard corpus. In contrast to previous tables, the combination schemes do not offer an improvement

Table 2. Performance of different combination schemes on various clustering algorithms for a 1000-document subset of Reuters-21578.

	Single Run	Best of 100 runs	SVD Combin.	Constr. Combin.	Unconstr. Combin.	No Combin.
$\boldsymbol{I_1}$						
Accuracy	.636	.644	.696	.669	.673	.686
NMI	.697	.697	.735	.711	.725	.726
$\boldsymbol{I_2}$						
Accuracy	.709	.797	.838	.838	.764	.808
NMI	.760	.805	.821	.819	.797	.814
$\boldsymbol{E_1}$						
Accuracy	.710	.797	.855	.837	.773	.849
NMI	.745	.790	.830	.819	.799	.822
$\boldsymbol{G_1}$						
Accuracy	.652	.660	.707	.721	.705	.709
NMI	.699	.716	.723	.727	.723	.727
$\boldsymbol{G_1'}$						
Accuracy	.692	.771	.814	.816	.782	.827
NMI	.730	.771	.797	.800	.790	.804
$\boldsymbol{H_1}$						
Accuracy	.709	.822	.844	.834	.789	.835
NMI	.758	.820	.821	.819	.801	.817
$\boldsymbol{H_2}$						
Accuracy	.719	.814	.854	.849	.799	.828
NMI	.761	.812	.837	.833	.813	.833
MixMulti						
Accuracy	.502	.525	.582	.543	.542	.586
NMI	.597	.609	.658	.644	.633	.651

for the CLUTO algorithms and for the unconstrained scheme there is even a degradation compared to the single run case. However, the mixture of multinomials records a very big improvement of about 40% on classification accuracy. It is interesting to note that for the Switchboard corpus, although the mixture of multinomials method was by far the worse clustering algorithm, after SVD combination it clearly became the best method. The same happened for the 20Newsgroups corpus where the mixture of multinomials was among one of the worse-performing methods and after SVD combination it became the best. These results suggest that when developing clustering algorithms, issues of the performance of metaclustering are distinct than issues of performance of single systems.

5.4 Factor analysis of results

In this subsection we try to determine the relative importance of two factors in the combination schemes: the mean and variance of the classification accuracy of individual systems. Comparing Table 1 or 2 with Table 3 the gains in 20News-

Table 3. Performance of different combination schemes on various clustering algorithms for Switchboard.

	Single Run	**Best of 100 runs**	**SVD Combin.**	**Constr. Combin.**	**Unconstr. Combin.**	**No Combin.**
$\boldsymbol{I_1}$						
Accuracy	.819	.848	.826	.820	.789	.836
NMI	.908	.914	.913	.907	.898	.915
$\boldsymbol{I_2}$						
Accuracy	.831	.863	.841	.837	.807	.845
NMI	.913	.920	.920	.918	.910	.922
$\boldsymbol{E_1}$						
Accuracy	.798	.819	.819	.777	.736	.818
NMI	.882	.886	.890	.883	.863	.891
$\boldsymbol{G_1}$						
Accuracy	.711	.711	.765	.751	.741	.762
NMI	.868	.870	.887	.877	.875	.888
$\boldsymbol{G_1'}$						
Accuracy	.789	.808	.811	.801	.749	.803
NMI	.875	.878	.880	.877	.859	.878
$\boldsymbol{H_1}$						
Accuracy	.826	.861	.842	.811	.757	.841
NMI	.910	.918	.918	.899	.895	.918
$\boldsymbol{H_2}$						
Accuracy	.814	.845	.840	.817	.773	.830
NMI	.897	.903	.905	.900	.886	.901
MixMulti						
Accuracy	.635	.699	.888	.756	.739	.876
NMI	.787	.818	.924	.899	.892	.921

groups or Reuters are higher than Switchboard and the variance of individual systems is higher in 20Newsgroups and Reuters than Switchboard. To assess the effect of each one of these two factors (mean and variance of individual systems) we generated 300 systems and chose a set of 100 for metaclustering depending on high/medium/low variance and similar mean (Table 4) or high/medium/low mean and similar variance (Table 5).

The results of Table 4 do not show a significant impact of variance on the combination results. The results of Table 5 show a clear impact of the mean on the combination results. However, from Tables 1, 2 and 3 we know that the performance of the combined system does not depend simply on the performance of the individual systems: the MixMulti result for Switchboard compared with the CLUTO results is a counterexample. It appears that there are unexplained interactions of mean, variance and clustering algorithms that will make the combination more successful in some cases and less successful in other cases.

Table 4. Effect of combining sets of 100 systems with approximately the same mean and different levels of variance. The (stdev,acc) cells contain the standard deviation and mean of classification accuracy for each set. Systems are generated with the E_1 criterion on 20Newsgroups and combined with SVD.

	Low Variance	**Medium Variance**	**High Variance**
(stdev,acc)	(.010,.577)	(.023,.578)	(.056,.580)
Accuracy	.640	.631	.635
NMI	.630	.629	.633

Table 5. Effect of combining sets of 100 systems with approximately the same variance and different levels of mean. The (stdev,acc) cells contain the standard deviation and mean of classification accuracy for each set. Systems are generated with the E_1 criterion on 20Newsgroups and combined with SVD.

	Low Mean	**Medium Mean**	**High Mean**
(stdev,acc)	(.018,.538)	(.010,.577)	(.019,.617)
Accuracy	.581	.641	.669
NMI	.616	.632	.647

6 Summary

We have presented three new methods for the combination of multiple clustering systems and evaluated them on three major corpora and on eight different clustering algorithms. Identifying the correspondence between clusters of different systems was achieved by "clustering clusters", using constrained or unconstrained clustering or by applying SVD. We have empirically demonstrated that the combination schemes can offer gains in most cases. Issues of combination of multiple runs of an algorithm can be important. The combination of different runs of mixture of multinomials algorithm was shown to outperform seven state-of-the-art clustering algorithms on two out of three corpora. In the future we will attempt to gain a better understanding of the conditions under which poor individual systems can lead to improved performance when combined.

References

1. Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286** (1999) 531–537
2. Schütze, H.: Automatic word sense discrimination. Computational Linguistics **24** (1998) 97–124
3. Zamir, O., Etzioni, O.: Grouper: a dynamic clustering interface to Web search results. Computer Networks **31** (1999) 1361–1374

4. Bellegarda, J.: Large vocabulary speech recognition with multispan statistical language models. IEEE Trans. on Speech and Audio Processing **8** (2000) 76–84
5. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31** (1999) 264–323
6. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning **36** (1999) 105–139
7. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Machine Learning Research **3** (2002) 583–617
8. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: a resampling-based method for class discovery and visualization of gene-expression microaray data. Machine Learning **52** (2003) 91–118
9. Fred, A., Jain, A.: Data clustering using evidence accumulation. In: Proc. of the International Conference on Pattern Recognition. (2002) 276–280
10. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics **19** (2003) 1090–1099
11. Zeng, Y., Tang, J., Garcia-Frias, J., Gao, G.: An adaptive meta-clustering approach: Combining the information from different clustering results. In: Proc. IEEE Computer Society Bioinformatics Conference. (2002) 276–281
12. Fern, X., Brodley, C.: Random projection for high dimensional data: A cluster ensemble approach. In: Proc. of the 20th International Conf. on Machine Learning, (ICML). (2003) 186–193
13. Topchy, A., Jain, A., Punch, W.: A mixture model for clustering ensembles. In: Proc. of SIAM Conference on Data Mining. (2004)
14. Dimitriadou, E., Weingessel, A., Hornik, K.: A combination scheme for fuzzy clustering. Inter. J. of Pattern Recognition and Artificial Intelligence **16** (2002) 901–912
15. Frossyniotis, D., Pertselakis, M., Stafylopatis, M.: A multi-clustering fusion algorithm. In: Proc. of the 2nd Hellenic Conference on Artificial Intelligence. (2002) 225–236
16. Bradley, P., Fayyad, U.: Refining initial points for K-Means clustering. In: Proc. 15th International Conf. on Machine Learning, (ICML). (1998) 91–99
17. Schwartz, G.: Estimating the dimension of a model. The Annals of Statistics **6(2)** (1978) 461–464
18. Godfrey, J., Holliman, E., McDaniel, J.: Switchboard: Telephone speech corpus for research development. In: Proc. of ICASSP. (1992) 517–520
19. Zhao, Y., Karypis, G.: Empirical and theoretical comparisons of selected criterion functions for document clustering. Machine Learning (2004) 311–331
20. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society **Series B, 39(1)** (1977) 1–38
21. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Learning to classify text from labeled and unlabeled documents. In: Proc. of AAAI. (1998) 792–799
22. Cheeseman, P., Stutz, J.: Bayesian classification (AutoClass): Theory and results. In: Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press (1996)