Clustering by intersection-merging*

Qunhua Li
Department of Statistics
University of Washington
Seattle, WA 98195
qli@stat.washington.edu

Marina Meilă
Department of Statistics
University of Washington
Seattle, WA 98195
mmp@stat.washington.edu

ABSTRACT

We propose Intersection-Merging (IM), a wrapper algorithm for model-based clustering. The algorithm takes a set of clusterings obtained e.g. by EM, breaks down the clusterings into subclusters via an intersection step, and then agglomerates them via a merging step. We introduce two versions of merging: greedy (standard IM) and by simulated annealing (IMSA). Experiments on several data sets show that both IM and IMSA improve on the starting clusterings under a variety of criteria.

1. INTRODUCTION

Model-based clustering and in particular clustering with mixtures of Gaussians using the EM [7] [2] algorithm are widely popular methods for clustering data. It is equally well-known [8] that these methods suffer from the presence of typically many local optima. It is therefore common practice to start the EM iteration from multiple initial points and to take the best solution (e.g. the one with maximum mixture likelihood). In this paper we propose to add an extra step that, instead of discarding all but one of the clusterings already obtained, combines them into a new one which will inherit strength from the whole ensemble. This is the Intersection-Merging algorithm.

Essentially, IM first partitions the data into smaller clusters, in a way that agrees with the inital clusterings, then agglomerates them using the model-based hierarchical approach.

The next section briefly describes model-based and hierarchical agglomerative clustering, then sections 3 and 4 introduce the IM and IMSA algorithms respectively. Experimental results are presented in section 5 and the discussion in section 7 concludes the paper.

9*A full version of this paper is available as *"Clustering by intersection-merging"* at `www.stat.washington.edu/spectral`

2. MODEL-BASED CLUSTERING AND MODEL-BASED HIERARCHICAL CLUSTERING

In model-based clustering, data are viewed as coming from a mixture density $f(x) = \sum_{k=1}^{K} \pi_k f_k(x)$. Here, π_k's are the *mixing proportions* ($0 < \pi_k \leq 1$ for all $k = 1, \ldots, K$ and $\sum_k \pi_k = 1$), and f_k is the density modeling *mixture component* k. A common assumption is that the densities f_k are multivariate Gaussian. The log-likelihood of the data for a Gaussian mixture with a given number K of mixture components is

$$l_{mix} = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \phi(x_i; \mu_k, \Sigma_k), \qquad (1)$$

where $\phi(\cdot; \mu_k, \Sigma_k)$ is the Gaussian density with mean vector μ_k and covariance matrix Σ_k and x_i, $i = 1, \ldots N$ is the i-th data point. For a fixed K we can estimate the parameters π_k, μ_k, and Σ_k using the EM algorithm [7]. The EM algorithm starts with an initial estimate for the parameters and iterates until convergence to a local maximum of the likelihood L_{mix}. Different initial points may result in different values for the parameters at convergencs. EM produces probabilistic assignments of points to clusters, aka *soft clusterings*. To obtain a *hard clustering*, point x_i is assigned to cluster k if

$$k = \text{argmax}_{k'} \pi_{k'} \phi(x_i; \mu_{k'}, \Sigma_{k'})$$

Throughout the paper, unless otherwise specified, a clustering will signify a hard clustering.

Hierarchical agglomerative clustering (HAC) [4] is another algorithm for model-based clustering. It starts with N clusters each containing one point and recursively merges two pair of clusters into a single one until K clusters are left. The two clusters for merging are chosen at each stage so as to maximize the resulting *classification (log-)likelihood*. The classification log-likelihood of a Gaussian mixture with a given number K of mixture components is

$$l_{cl} = \sum_{i=1}^{N} \log \phi(x_i; \mu_{l_i}, \Sigma_{l_i}), \qquad (2)$$

where the classification label $l_i = k$ if x_i belongs to the kth component.

Whereas EM search is local and its movement is in the neighborhood of the initial point in parameter space, HAC is not influenced by initialization. It changes the assignment on a

large scale via agglomeration. But it is a greedy algorithm. Once observations from different groups have been assigned to the same cluster this error will never be corrected. Also, the calculation of HAC is expensive in both memory usage and computing time ($O(N^3)$ for the worst case).

Both algorithms assume that the value of K is known and arrive at local maxima of their respective optimality criteria. The location of the optima is not the same for the two algorithms, but they are close to each other when the mixture components are well separated.

Throughout this paper, we assume that the number of clusters (K) is known and will focus on estimating the parameters that maximize the mixture likelihood.

3. THE INTERSECTION-MERGING ALGORITHM

ALGORITHM INTERSECTION-MERGING

Input Data $x_1, \ldots x_N$, *starting clusterings* $\mathbb{C} = (C_1, \ldots, C_n)$

1. **Intersect (I)** starting clusterings to produce clustering C_r with K_r *subclusters*
2. **Merge (M)** subclusters into K clusters using HAC
3. **Locally optimize** by EM initialized with the parameters from the mixture model obtained in step 2

The algorithm is illustrated as above. The starting clusterings are obtained for example from EM with different initializations. The **I** step produces a partition C_r of the data into K_r clusters such that x_i, x_j are in the same cluster in C_r iff they are in the same cluster for each of the starting clusterings C_l, $l = 1, \ldots n$. The clusters of C_r are called *subclusters* to emphasize the fact that each of them is a subset of a cluster of C_l for every $l = 1, \ldots n$. If the starting clusterings all capture to some extent some true clustering of the data, the data points with the same true membership are more likely to be kept in the same subcluster as pairs of data points that belong to different true clusters. However, the subclusters generated by the intersection are just substructures of clusters. The merging step is to regroup them into clusters by agglomeration. When the agglomeration is completed, it is optimized locally using EM.

It is important to note that the number of subclusters K_r does not grow exponeitially with the increase of K and n. In the worst case K_r attains the size of the dataset N, in which case our algorithm reduces to standard HAC. Usually K_r is much smaller than N. Figure 5 a. shows the distribution of K_r in 100 datasets generated from the distribution in figure 1.

In this paper, we introduce an $O(N)$ time implementation for the intersection step using hashing and a straightforward implementation for hierarchical model-based clustering which is $O(K_r^3)$ in the worst case, but typically $O(K_r^2)$ [4].

4. INTERSECTION-MERGING WITH SIMULATED ANNEALING

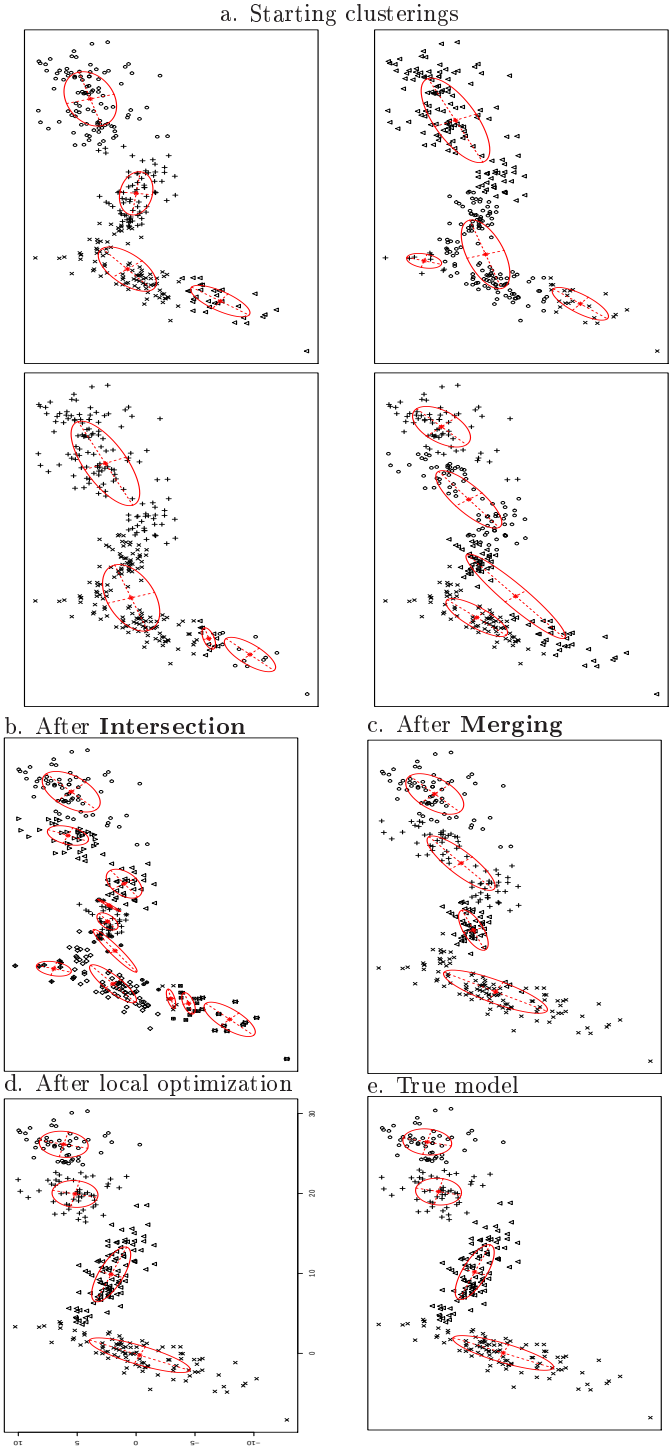


Figure 1: An illustration of the IM algorithm for a data set with 4 clusters. Clusters centers and 1 standard deviation shown in red. (a) Four starting clusterings generated from EM; none of them is close to the true clustering. (b) the intersection of starting clusterings has $K_r = 12$ subclusers; (c) after the Merging step (as IM-agg); (d) after the final EM optimization (as IM-EM) the obtained clustering and mixture model is very close to the truth (e).

In the IM algorithm described above, the agglomeration in the merging step is greedy. Hence, once observations from different groups have been assigned to the same cluster this error will never be corrected. Therefore, we propose an enhanced version of IM, intersection-merging with simulated annealing (IMSA), which allows the correction of mistakes formed in agglomeration by a procedure of simulated annealing.

ALGORITHM INTERSECTION-MERGING WITH SIMULATED ANNEALING (IMSA)

Input Data $x_1, \ldots x_N$, starting clusterings $\mathbb{C} = (C_1, \ldots, C_n)$ Parameters n_{iter}, m, T

1. Repeat n_{iter} times
   (a) Randomly select a subset of size m, $(C_{k_1}, \ldots, C_{k_m}) \subseteq \mathbb{C}$
   (b) Apply Intersection and Merging on $(C_{k_1}, \ldots, C_{k_m})$, get clustering C_{new}
   (c) Calculate the difference of classification likelihood as $\Delta l_{cl} = l_{C_{new}} - \max_{1 \leqslant i \leqslant m} l_{C_{k_i}}$
   (d) **If** $\Delta l_{cl} > 0$, ACCEPT and set $C^* = \text{argmax}_i l_{C_i}$ **else** ACCEPT with probability $\exp(\Delta l_{cl}/T)$.
2. **Locally optimize** by EM initialized with the parameters from the mixture model corresponding to C^*

ACCEPT: Let $l = \underset{1 \leqslant i \leqslant m}{\text{argmin}}\, d(C_{k_i}, C_{new})$. Replace C_l by C_{new}.

The above algorithm is an instance of the Metropolis-Hastings simulated annealing algorithm [11], where: the current configuration is the current set of clusterings, $(C_1, \ldots, C_n)$, the proposal is the new clustering C_{new} generated by applying steps I and M on the subset $(C_{k_1}, \ldots, C_{k_m})$, and the objective function is the classification log-likelihood summed over the configuration $\Sigma_{\mathbb{C}} l_{C_i}$.

There are many possibilities for the annealing schedule. The one we use here is to select a temperature parameter T such that the acceptance rate is reasonably far from 0 and 1. This value of T is selected in a set of small trial runs whose results are discarded; once selected, T is constant throughout the annealing process. We have found this simple schedule to work sufficiently well in our experiments.

There can be many variants as to the rearrangement of the configuration. We accept the proposal when the proposed clustering is better than the best clustering in the subset with probability 1; otherwise, replace the clustering in the subset closest to C_{new} by C_{new} with probability $\exp(\Delta l_{cl}/T)$. The cluster distance is measured by variation of information (VI) [9]. This particular choice helps preserve the diversity of the clusterings in the configuration. Our experiments showed that perserving diversity of $\mathbb{C}$ is essential to avoid early convergence to a configuration with essentially identical clusterings. If diversity diminishes too soon, intersection also creates many tiny subclusters, which significantly increase the error rate in merging and introduce large overheads. The size of the subset m in general should not be too small or too large to be close to the size of starting clusterings n, because an extremely large m will reduce the randomness of the subset and an extremely small m will limit the change of $\mathbb{C}_{new}$.

5. EXPERIMENTS AND RESULTS

All the models are implemented in MCLUST [5]. For a Gaussian mixture, the covariance matrix for the kth cluster can be expressed in the form

$$\Sigma_k = \lambda_k D_k A_k D_k^T, \tag{3}$$

where λ_k is a scalar determining the volume of the cluster, D_k is the matrix of eigenvectors determining the orientation, and A_k is a diagnal matrix proportional to the eigenvalues determining the shape of the cluster. Banfield and Raftery proposed cross-cluster equality constraints on any or all of these geometric features as a way of limiting the number of parameters in the model in an intuitive way [2]. For example, one such model constrains all clusters to have the same shape, volumes and orientation. This is called the EEE model (Equal volume, Equal shape and Equal orientation) [5]. A completely unconstrained model is denoted by VVV [5]. A discussion of all possible combinations of constraints based on the decomposition can be found in [3]. We found that on our datasets, the VVV variant of MCLUST converged most of the time to singularities (i.e clusters with 1 data point). As this was impractical for producing any reasonable of starting clusterings, we adopted the following experimental design: we obtained the starting clusterings with the more stable EEE, applied IM/IMSA, them EM-VVV for the final step of our algorithm. We also applied EM-VVV to the n starting clusterings, and these are the results we report for them.

5.1 A Simple Mixture

We first evaluate the performance using the demonstration example in Section 3. The dataset contains 300 data points from 4 groups with $\mu_1 = (10, 2)$, $\mu_2 = (0, 0)$, $\mu_3 = (20, 5)$, $\mu_4 = (26, 6)$,

$$\Sigma_1 = \begin{pmatrix} 16 & -6.4 \\ -6.4 & 4 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 4 & 6.4 \\ 6.4 & 16 \end{pmatrix}, \Sigma_3 = \Sigma_4 = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix},$$

and the mixing proportion $\pi_1 = \frac{1}{3}$, $\pi_2 = \frac{1}{3}$, $\pi_3 = \frac{1}{6}$, $\pi_4 = \frac{1}{6}$. Figure 1, e shows one realization of the distribution. The parameters are chosen such that the two spherical groups (C_3 and C_4) are easily to be mixed up as a big ellipse.

We generate 20 random samples of size $N = 300$ from the mixture distribution above. Then we initialize EM in the follow ways. For each sample, we randomly generate the starting mean $\mu_i^{(0)}$ independently as

$$\mu_1^{(0)}, \ldots, \mu_K^{(0)} \sim N(\bar{x}, V),$$

where $\bar{x}$ is the sample mean and V is the sample covariance matrix of the sample. Then we specify the component-covariance matrices $\Sigma_i^{(0)} = V$ and $\pi_i^{(0)} = 1/K$ to generate 100 partitions using EM (EEE model, i.e. spherical shape equal covariance) with MCLUST[5]. A browsing of the partitions shows that there is substantial redundancy in the

100 clusterings. The redundant clusterings increase the size of intersection and produce many tiny subclusters, which consequently increase computing overhead and the chance of making an error in the agglomeration. To preserve the maximum diversity of partitions in the reduction of redundancy, we use the anchor algorithm [10] with variation of information [9] as the distance measure to select the 10 most diversified partitions as the effective starting clusterings for IM or IMSA.

There are several adjustable parameters in IMSA. Here, we use 10 starting clusterings ($n = 10$) and choose a subset of size 3 ($m = 3$) for each iteration of IM. The updating strategy is exactly as described in Section 4. It is a scheme combining diversity and likelihood, i.e. one branch is updated by likelihood, and the other is by closeness of clusterings (See Section 4). This design is to avoid losing diversity of clusterings in $\mathbb{C}$ rapidly, which may lead to going downhill too quickly and stopping moving too soon. In the experiments, we use $T = 10$ and keep T constant throughout the annealing. We set up iterations=1000 and keep track of the clustering with the highest classification likelihood in each iteration. The grand best clustering will be the agglomeration result of IMSA. In all the experiments under current setup, we observe that the likelihood of the best clustering increases with the iteration. We define convergence as when the likelihood of the best clustering does not change. The convergence may not be achieved when it stops.

A final EM optimization with the VVV model (i.e. unconstrained covariance) is applied to the agglomeration results at the end. For clarity, we use IM-agg (or IMSA-agg) to refer the results before the final EM step and IM-EM (or IMSA-EM) for the results after the final EM step. For comparison purposes, the same EM procedure is applied to the best starting clusterings (i.e. the starting clusterings with highest mixture likelihood in the pool) and the true labels, denoted as bestVVV and emtruth, respectively.

Figure 2 summarizes the results of IM and IMSA in terms of the mixture likelihood and classification likelihood for the 20 datasets. All IMSA-agg results and 44% of IM-agg results have higher classfication likelihood than the best starting clusterings. All IMSA-agg results have higher or the same classification likelihoods comparing to IM-agg results. Both IM-agg and IMSA-agg have higher mixture likelihoods than the best starting clustering. This indicates that IM-agg and IMSA-agg improve likelihood of clusterings and the simulated annealing does improve agglomeration. After EM (VVV model) optimization, IM-EM, IMSA-EM and bestVVV have similar mixture likelihood. Both IM-EM and IMSA-EM reach the optimal mixture likelihood (emTruth) 35% of the time.

To assess the quality of partitions, we evaluate the classification error (CE), variation of information (VI) and mixture likelihood on the training sets and a test set (Figure 3). The test set is generated from the same mixture distribution containing 3000 data points. Comparing to the best starting clustering (bestVVV), both IM-EM and IMSA-EM reduce the average classification errors on both the training sets and the test set. They also shortern the average VI distance of the estimated clustering to true clustering

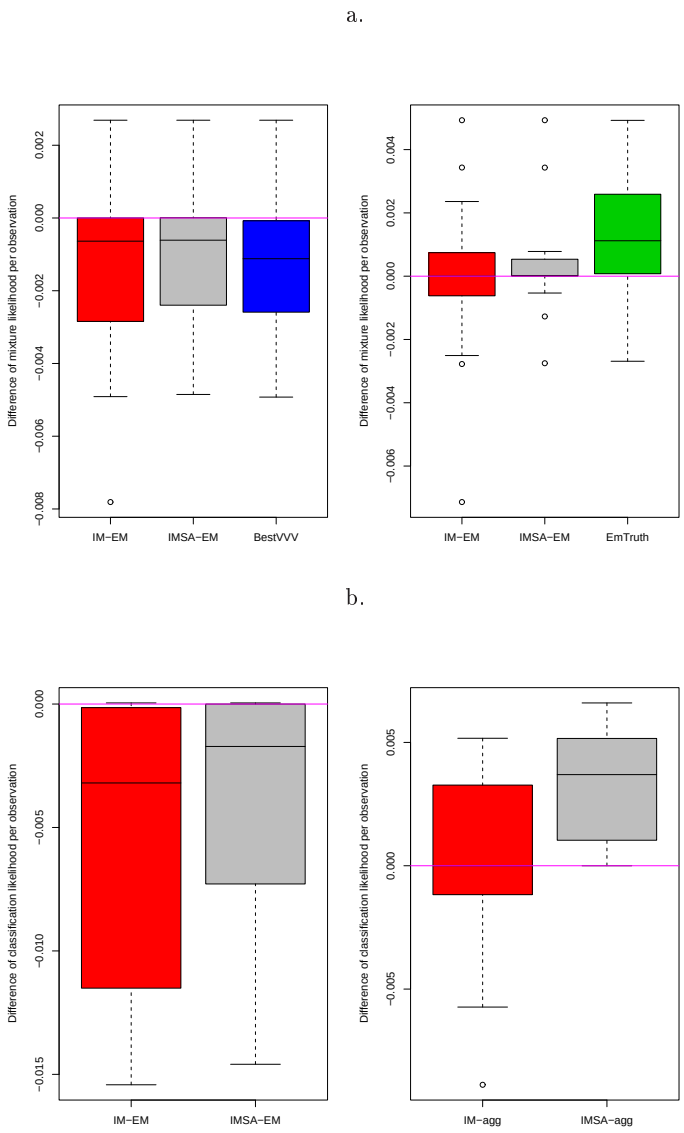


Figure 2: Mixture likelihood and classification likelihood for the 4-component mixture. (a) Difference of mixture likelihood to emTruth (left) and bestVVV (right). (b) Difference of classification likelihood to emTruth (left) and best starting clustering (right)

on both the training sets and the test set. IM, IMSA and bestVVV have similar mixture and classification likelihood per observation. (Table 1).

5.2 Duplicated data

In section 5.1, the mixture likelihood of the optimal clustering and the best starting clusterings are fairly close. Now we want to find a harder example. A natural choice is to duplicate the data in section 5.1. We keep the covariances and relative proportions of $C_1, \ldots, C_4$ in section 5.1, and make a shifted duplicate at $(17, -11), (26, -9), (0, -15)and(6, -14)$. So we end up with a dataset containing 600 points from 8 groups.

Table 1: Simple mixture data. Classification error (CE), variation of information (VI) distance to the true labeling, log-mixture likelihood per observation (l_{mix}/N) and log-classification likelihood per observation (l_{class}/N) on the training sets and the test set. The table shows the means and standard deviations over the estimations obtained from the 20 training datasets. The test set is generated from the same mixture distribution with 3000 data points.

	Training				Test			
	CE	VI	l_{mix}/N	l_{class}/N	CE	VI	l_{mix}/N	l_{Class}/N
IM	0.185 (0.117)	0.794 (0.254)	-8.013 (0.075)	-6.228 (0.096)	0.236 (0.143)	1.074 (0.353)	-8.097 (0.040)	-6.427 (0.199)
IMSA (VI)	0.195 (0.126)	0.801 (0.234)	-8.010 (0.080)	-6.228 (0.131)	0.220 (0.114)	1.033 (0.249)	-8.088 (0.029)	-6.445 (0.176)
BestVVV	0.218 (0.107)	0.843 (0.211)	-8.013 (0.083)	-6.256 (0.138)	0.255 (0.096)	1.088 (0.238)	-8.093 (0.028)	-6.502 (0.182)
EmTruth	0.056 (0.019)	0.580 (0.135)	-7.992 (0.073)	-6.143 (0.073)	0.123 (0.045)	0.677 (0.067)	-8.071 (0.031)	-6.313 (0.100)

We use the same IM and IMSA procedures as before except the effective starting clusterings $n = 20$ and subset size $m = 4$ in IMSA as the starting clusterings are more diversified with larger number of clusters. Figure 4 shows the classification likelihood and mixture likelihood. The mixture likelihoods of IM-EM and IMSA-EM are better than the mixture likelihood of the best clustering after the EM step most of time (IM: 90% and IMSA: 80%), which is better than the 4-component mixture. The classification and the mixture likelihood of IMSA-agg are always higher than IM-agg and the best starting clustering. However, IM sometimes has higher mixture likelihood than IMSA after the final EM step.

Similar to the 4-component mixture, we generate a test set which is 10 times the size of the training sets. Both IM-EM and IMSA-EM have lower means of classification error, variation of information and KL distance than the best starting clustering (bestVVV) on both the training sets and the test set (Table 2).

5.3 Digit data

We test our algorithms on the optical handwritten digit recognition that is available in the NIST site [1]. The description of the data and preprocessing can be found in [1]. The dataset contains the counts of pixels in small squares of bitmap images of handwritten digits 0-9 collected from 43 people, in which 30 people contributed to the training set (3823) and 13 to the test set (1797). It is an 8×8 matrix of integer in the range of $0 \ldots 16$ for each digit. We obtain 10 training sets by randomly sampling 100 datapoints per digit. Each training set contains 1000 64-dimension datapoints with 10 clusters which we further project to 11 dimensions by a random orthogonal matrix. We use the test set provided in the dataset as the test set. The results of IM and IMSA are in table 3.

On the training sets IM has higher mixture likelihood and classification likelihood than BestVVV, but larger classification error and VI distance. All the methods have fairly high classification error on the test set. We observe that sometimes the number of clusters is lower than the prespecified value (10) on the test set, which contributes to the high classification error. Also, some small clusters (≤ 10) generated due to the same reason result to the occurrence of singularity on test set.

6. RELATED WORK

Several other approaches have been proposed to improve EM by turning to agglomerative variants. Frigui and Krishnapuram proposed a Competitive Agglomeration algorithm that starts with a large number of small clusters and have adjacent clusters compete for data points iteratively [6]. As the clusters that lose the competition gradually become depleted and vanish, they obtain a sequence of partitions with progressively diminishing number of clusters. The final partition is taken to have the optimal number of clusters based on an objective function that inherits the advantages of hierarchical clustering. Shoham proposed a deterministic annealing EM algorithm in agglomeration mode by combining deterministic annealing and competitive agglomeration [13].

A number of approaches have been proposed to improve initial clusterings, especailly on the application on large datasets. Posse proposed to improve hierarchical agglomeration from an efficient classification of the data in many classes rather than from the usual set of singleton clusters by using minimum spanning tree [12]. Tantrum et al clustered the fractions of data splitted through refraction and clustered the meta-observations summarized by their means [14].

7. DISCUSSION

In this paper, we have presented a new idea of clustering using intersection-merging. We introduced two versions of merging: greedy (IM) and by simulated annealing (IMSA). We have shown that IM and IMSA improve on the initial clustering under a variety of criteria.

In our experiments, improvement is more significant on datasets with more clusters. This probably is because it is more difficult to have completely correct starting clusterings in this situation, but IM or IMSA can combine the merits of partially correct starting clusterings and produce a better product.

We also observe that IMSA can correct mistakes during agglomeration. Thus, the classification likelihood of IMSA-agg is consistently higher than IM-agg. The difference is more significant when the data contains more clusters. This can be explained by the increase in the size of the intersection with the increase in the number of clusters. When the number of starting clusterings (n) or the number of components (K) increases, the intersection size, K_r, increases rapidly. In this situation, the chance of making an error in agglomeration increases for IM. However, in IMSA K_r does not depend on n but on the number of clusterings intersected

Table 2: Duplicated mixture data. Classification error (CE), variation of information (VI) distance to the true labeling, log-mixture likelihood per observation (l_{mix}/N) and log-classification likelihood per observation (l_{class}/N) on the training sets and the test set. The table shows the means and standard deviations over the estimations obtained from the 20 training datasets. The test set is generated from the same mixture distribution with 6000 data points.

	Training				Test			
	CE	VI	l_{mix}/N	l_{class}/N	CE	VI	l_{mix}/N	l_{class}/N
IM	0.191 (0.051)	0.821 (0.117)	-8.956 (0.047)	-6.149 (0.071)	0.232 (0.045)	1.084 (0.103)	-9.067 (0.019)	-6.362 (0.044)
IMSA (VI)	0.214 (0.060)	0.865 (0.125)	-8.964 (0.047)	-6.158 (0.056)	0.236 (0.085)	1.006 (0.188)	-9.066 (0.014)	-6.351 (0.086)
BestVVV	0.278 (0.059)	0.964 (0.112)	-8.946 (0.047)	-6.200 (0.060)	0.317 (0.069)	1.102 (0.190)	-9.075 (0.021)	-6.422 (0.096)
EmTruth	0.059 (0.013)	0.631 (0.110)	-8.977 (0.047)	-6.100 (0.050)	0.074 (0.006)	0.727 (0.033)	-9.041 (0.014)	-6.227 (0.016)

Table 3: Handwritten digits data. Classification error (CE), variation of information (VI) distance to the true labeling, log-mixture likelihood per observation (l_{mix}/N) and log-classification likelihood per observation (l_{class}/N) on the training sets and the test set. The table shows the means and standard deviations over the estimations obtained from the 10 training datasets. Test set is sampled from the original test set. It contains 10 digits with 100 observations from each digit. Due to singularity, classification likelihood for the test set is not available. The criteria for IM are averaged over 9 non-singular models.

	Training				Test		
	CE	VI	l_{mix}/N	l_{class}/N	CE	VI	l_{mix}/N
IM	0.371 (0.027)	2.435 (0.161)	-39.177 (0.102)	-36.046 (0.140)	0.681 (0.141)	3.316 (0.126)	-41.531 (0.1097)
IMSA	0.318 (0.039)	2.350 (0.170)	-39.110 (0.107)	-7.446 (0.119)	0.716 (0.161)	3.282 (0.130)	-41.383 (0.132)
IMSA-VI	0.330 (0.065)	2.315 (0.246)	-39.154 (0.077)	-7.450 (0.075)	0.638 (0.160)	3.233 (0.252)	-41.487 (0.197)
BestVVV	0.378 (0.048)	2.406 (0.195)	-39.267 (0.112)	-36.169 (0.082)	0.580 (0.168)	3.083 (0.348)	-41.614 (0.164)
EmTruth	0.176 (0.033)	1.709 (0.183)	-39.208 (0.123)	-35.950 (0.088)	0.664 (0.052)	3.127 (0.152)	-41.562 (0.143)

at each stage m, which is much smaller; IMSA is also less sensitive to to the increases in intersection size, because it can break down the incorrect agglomeration in later iterations. It may be surprising then that IMSA does not dominate IM also after the final EM relaxation. In fact, IM and IMSA have similar likelihood . This is explained by the fact that, for the experiments in this paper, we chose data with partly overlapping clusters, for which the parameters that optimize l_{mix} and l_{cl} differ significantly enough that the apparent advantage of IMSA does not show in the final value of the mixture likelihood.

The diversity of starting clusterings is also essential to the performance of both IM and IMSA. One can easily see that, the final EM optimization left aside, neither IM or IMSA can find a clustering unless it can be obtained by merging of subclusters in the intersection. Thus if the starting clusterings are all similar, so will be the final result of IM/IMSA.

When the starting clusterings have similar partitions, the subclusters generated during the intersection are either close to the clusters in the starting clusterings or small subclusters reflecting the difference between starting clusterings. Merging those subclusters usually will not lead to improvement but will instead introduce noise. If this happens, one possibility is that the pool of starting clusterings is not large enough. Increasing the pool may increase the diversity. If the pool is already large enough and all starting clusterings are similar, then this may be a happy case that EM has found global optima. This is also a case where we do not expect significant improvement from IM. On the other hand, if the starting clusterings are too diversified that the number of subclusters are close to data size N, IM is degenerated to a hierarchical agglomerative clustering. Thus IM is most useful in the middle ground. A typical favorable case for IM is when many subclusters after intersection have moderate sizes. This suggests a useful diagnosis procedure: the distribution of subcluster sizes after the I step can predict (quite reliably as we have observed in our experiments) if IM will be able to improve on the starting clusterings or not.

Note also that, since except for the final relaxation step IM/IMSA work with hard clusterings, when the mixture components overlap too much our algorithms cease to be useful. Thus, unlike EM, they are not general purpose density estimation algorithms, but clustering algorithms. Also, when the clusters are well separated, EM usually finds the optimal parameters and partition, so IM has no room to improve. It may seem thus that our new algorithms have restricted applicability. However, in our opinion this is not so: it is in the intermediate domain where clusters exist in the data but they are not easy to find that clustering is really interesting. This is exactly the domain where IM/IMSA apply. What amount of cluster overlap can be tolerated by IM? We do not have theoretical results, but in the artificial experiments presented here, the overlap of the small clusters is 30% while in the digits data the maximum overlap in the true clustering is

In practice, a common situation is a big pool with several types of partitions. Our practical solution is to use the anchor algorithm [10] with variation of information [9] as the distance measure to select the most diversified starting clusterings. For the reasons expounded here we strongly recommend selecting the set of starting clusterings by diversity as a preprocessing step for IM.

In our current algorithm, the selection criterion in IMSA

is classification likelihood, but the final EM optimization is based on the mixture likelihood. Since there is no strict correspondence between mixture likelihood and classification likelihood, the optima from IMSA may not be close to the optima on the mixture likelihood. Therefore, we also experimented using the mixture likelihood as the objective function of simulated annealing. The results so far have been similar to the standard IMSA; we do not yet have a good explanation for the lack of improvement. We are currectly developing an Intersection-Merging algorithm based entirely on mixture likelihood so that all steps optimize the same objective function.

There are several possible extensions of the work presented here. In this paper, we assume the number of components K is known and all the starting clusterings have the same K. Actually this is not required for IM or IMSA. We may even select the optimal K by running IM/IMSA under different K and making selection based on for example BIC. So far we applied the IM or IMSA framework on Gaussian mixtures. But conceptually this framework can apply to any mixture problems, as long as some starting clusterings and an optimality function for agglomeration are available. Another possible application is to obtain hierarchical clusterings on large datasets. For large datasets, HAC on the whole dataset is infeasible in terms of memory usage and computation time. We may use EM (perhaps on samples of the dataset) to obtain starting clusterings and produce the clustering on the full dataset using Intersection-merging whose agglomeration step depends on K_r but not on N.

8. ACKNOWLEDGEMENT

This was partically supported by UW Royalty Research Fund Grant 65-1870 and NSF Grant IIS-0313339.

9. REFERENCES

[1] Optical recognition of handwritten digits.

[2] J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49, 1993.

[3] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28, 1995.

[4] C. Fraley and A. E. Raftery. How many clusters? which clustering method? - answers via model-based cluster analysis. *The Computer Journal*, 41, 1998.

[5] C. Fraley and A. E. Raftery. Mclust: Software for model-based clustering, discriminant analysis, and density estimation. Technical Report 415, Department of Statistics, University of Washington, Oct. 2002.

[6] H. Frigui and R. Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30, 1997.

[7] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

[8] G. J. McLachlan and T. Krishnan. *The EM algorithm and Extensions*. John Wiley & Sons, 1997.

[9] M. Meila. Comparing clusterings. Technical Report 418, Department of Statistics, University of Washington, 2002.

[10] A. Moore. The anchors hierarchy: Using the triangle inequality to survive high-dimensional data. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*. AAAI Press, 2000.

[11] R. Otten and L. van Ginneken. *The Annealing Algorithm*. Kluwer, 1989.

[12] C. Posse. Hierarchical model-based clustering for large datasets. *Journal of Computational and Graphical Statistics*, 10(3), 2001.

[13] S. Shoham. Robust clustering by deterministic agglomeration em of mixture of multivariate t-distributions. *Pattern Recognition*, 35, 2002.

[14] J. Tantrum, A. Murua, and W. Stuetzle. Hierarchical model-based clustering of large datasets through fractionation and refractionation. In *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2002.

APPENDIX

A. IMPLEMENTATION OF INTERSECTION

We developed a linear time algorithm for the intersection step We call the vector of cluster labels in the input partitions for each data point as a tuple. The data points in the same subcluster will have the same tuple, i.e. each tuple uniquely identifies an intersection (which bears a unique cluster label). A hashtable data structure, which provides a one-to-one mapping between keys (tuple) and values (index in the intersection), is used to implement the intersection process. The intersection will be formed after a single traverse of the whole data set, i.e. O(N).

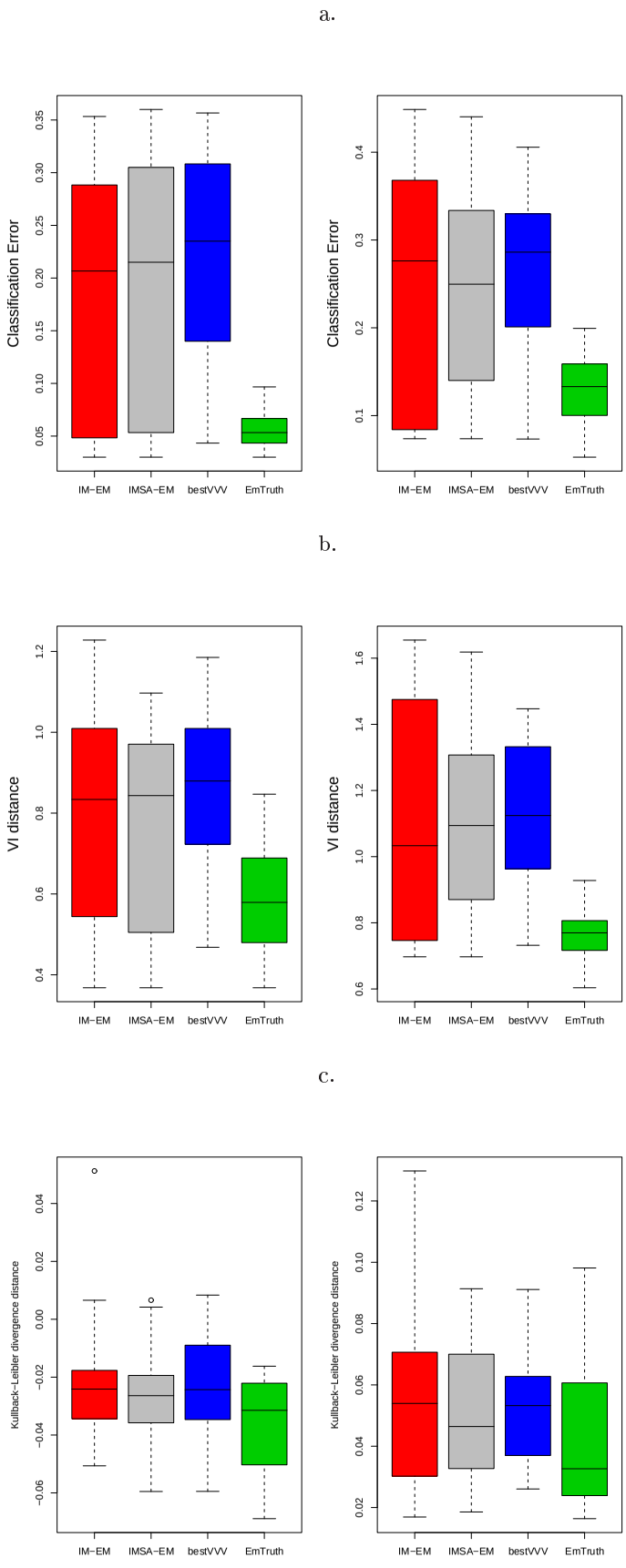


Figure 3: Classification error, variation of information and mixture likelihood for the 4-component mixture. Left: training sets. Right: test set.

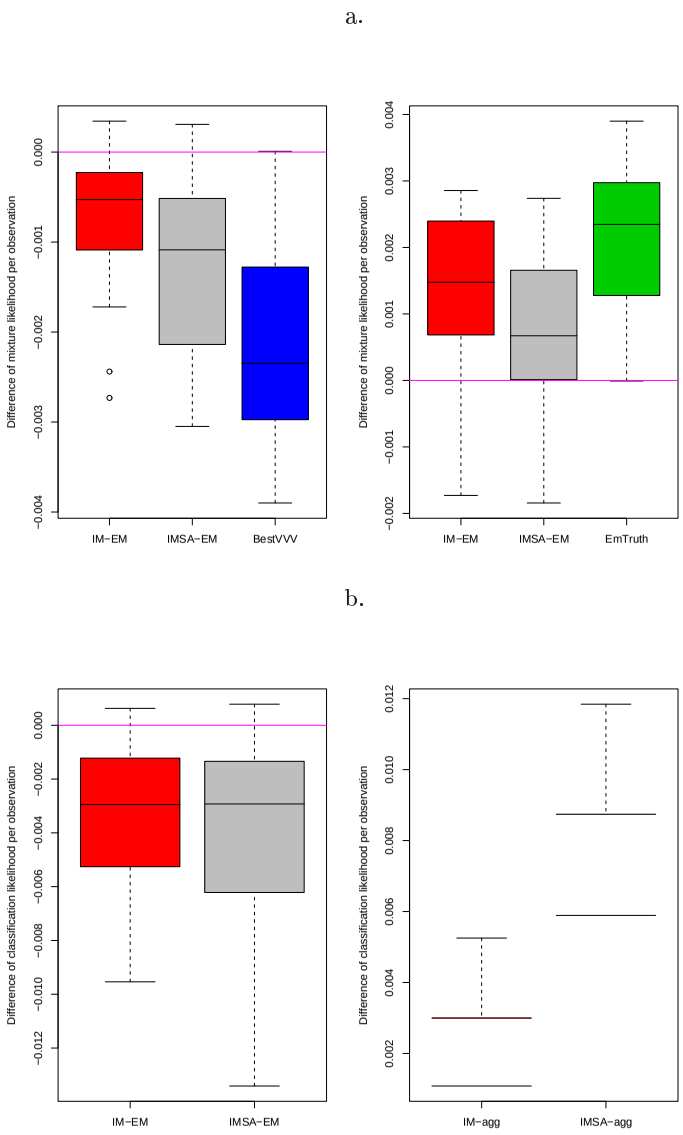
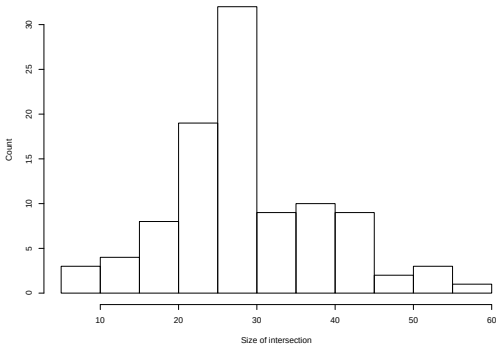


Figure 4: Mixture likelihood and classification likelihood for the 8 components mixture. (a) Difference of mixture likelihood to emTruth (left) and bestVVV (right). (b) Difference of classification likelihood to emTruth (left) and best starting clustering (right). Due to singularity in the calculation, only 3 classification likelihood is available for IM-agg and IMSA-agg.

a. artificial data $K = 4, n = 10, N = 300$



b. artificial data $K = 8, n = 20, N = 600$

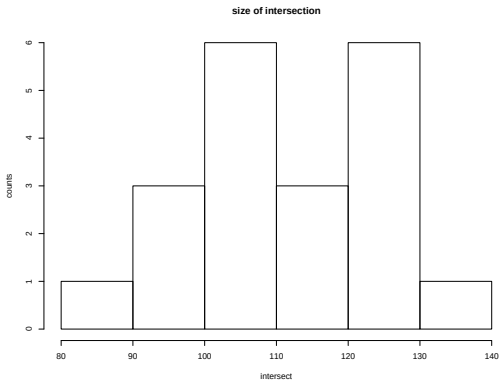


Figure 5: The distribution of size of intersection K_r . The median size of intersection is 28 for the 4-component dataset (a), and 111 for the 8-component dataset (b).