Consensus Clustering for Detection of Overlapping Clusters in Microarray Data

Meghana Deodhar and Joydeep Ghosh
Department of Electrical and Computer Engineering
University of Texas at Austin, TX, USA
deodhar, ghosh@ece.utexas.edu

Abstract

Most clustering algorithms are partitional in nature, assigning each data point to exactly one cluster. However, several real world datasets have inherently overlapping clusters in which a single data point can belong entirely to more than one cluster. This is often the case with gene microarray data since it is possible for a single gene to participate in more than one biological process. This paper deals with a novel application of consensus clustering for detecting overlapping clusters. Our approach takes advantage of the fact that results obtained by applying different clustering algorithms to the same dataset could be different and a consensus across these results could be used to detect overlapping clusters. Moreover we extend a popular model selection approach called X-means [10] to detect the inherent number of overlapping clusters in the data.

1 Introduction

Several real world domains such as bioinformatics, text clustering and recommender systems involve data with inherently overlapping clusters, where some items belong entirely to two or more clusters [1, 11]. This is different from soft or fuzzy clustering where an object can be assigned to multiple clusters with different probabilities or degrees of membership. This paper deals with clustering microarray data, where genes could have more than one function and could be part of multiple metabolic pathways and hence multiple clusters. Clustering algorithms that allow membership to more than one cluster are more suitable for modeling the underlying structure of such datasets. The approach we have developed to detect overlapping clusters is based on consensus clustering, which involves combining the results of multiple base clustering algorithms to obtain the final clustering. Existing works on consensus clustering (also called cluster ensembles) are motivated by the possibility of knowledge reuse [12], improved quality and robustness of results [13], etc. and this paper presents a novel application of this approach. Specifically, we propose both a discriminative and a generative approach to the problem of using the results of consensus clustering to obtain overlapping clusters in section 4. A model selection algorithm for detecting the natural number of overlapping clusters in the data is also developed. These algorithms are shown to be quite effective on microarray data in section 5.

2 Problem Definition

We assume that the individual clustering solutions, i.e. the base clusterings that are used to form the cluster ensemble are hard partitional, where a data point is assigned to exactly one cluster. The aim in this paper is to develop a consensus algorithm that uses such base results to perform a clustering that allows overlapping clusters. The intuition behind such a consensus algorithm is that if a data point belongs completely to two (or more) clusters, the base clustering algorithms will be forced to assign the point to one of the two clusters. In the ideal case half the base clustering algorithms will assign the point to one of the clusters and the other half will assign it to the other cluster. A consensus across these results will assign the gene a high membership to both the clusters, hence it will be assigned to both. For this approach to work, it is necessary for the base clustering solutions to be diverse, but still of good quality. These desiderata are similar to those for regression or classification ensembles [14]. The inputs to the overlapping cluster detection algorithm are the dataset along with the sets of cluster labels assigned to the data by the base clustering algorithms. The final output of the algorithm is cluster memberships, assigned to each data point, where a point may be a member of more than one cluster.

3 Related Work

The problem of combining multiple partitionings of a set of objects into a single consolidated clustering has been addressed by Strehl et al. [12]. They pose the consensus clustering problem as finding a consensus function that combines clustering results from a variety of sources without using the original features or algorithms. Three computationally efficient heuristics for combining multiple clusterings are proposed, each of which find an optimal clustering that maximizes the average normalized mutual information (ANMI) with the individual clusterings.

Consensus clustering has recently been applied to microarray data to improve the quality and robustness of the resulting clusters. A resampling based approach is used by Monti et al. [9], in which the agreement across the results obtained by executing a base clustering algorithm on several perturbations of the original dataset is used to obtain the final clustering. Swift et al. [13] use a variety of clustering algorithms on the same dataset to generate different base clustering results and try to find clusters that are consistent across all the base results using simulated annealing. Another formulation of the consensus clustering problem is as a median partition problem [4], where the aim is to find a partitioning of the data points that minimizes the distance to all the other partitionings. The authors propose greedy heuristic solutions to find a local optimum. However, none of these approaches take overlapping clusters into account and find only disjoint clusters.

Once it was realized that overlapping clusterings were natural in several domains including bioinformatics, several researchers devised ways of attaining such solutions outside of the consensus framework. For example, a model for overlapping clustering, particularly in the context of microarray data was introduced by Segal et al. [11], which is essentially the specialization of a Probabilistic Relational Model (PRM). Banerjee et al. present an alternative and simpler view of this model as a mixture model of exponential density functions [1]. The main advantage of this new model is that it is generalized to work with any Bregman divergence, making it applicable to a wide range of distance functions.

4 Overlapping Clustering Algorithms

As is the case with classification, one can characterize clustering approaches as being discriminative or generative (probabilistic or model based) [7]. We propose one approach from each category in this section. Note that for the overlapping clustering problem, neither viewpoints can fully avoid the choice of a "threshold" be it distance or affinity based or probability based, that governs the amount of overlap indicated in the solution. So an important issue is to find approaches that are relatively insensitive to the choice of such a threshold.

Discriminative Approach: Thresholded MCLA. Of the various discriminative approaches to consensus clustering, we saw that the Meta-Clustering Algorithm (MCLA) [12] offered the most natural extension for generating overlaps. The modified thresholded MCLA algorithm outlined below can be used to detect overlapping clusters.

1. Represent each base cluster by a hyperedge, transforming the base clustering results into a hypergraph. Each hyperedge has an indicator vector h associated with it, with one entry for each data point such that $h_i = 1$ if data point i belongs to the cluster, 0 otherwise.

2. Cluster related hyperedges into meta-clusters using the hypergraph partitioning package of METIS [8].

3. Build an association vector m for each meta-cluster which is computed by taking an element wise average of the indicator vectors of the hyperedges belonging to the meta-cluster. The association vector has an entry for each data point which describes the level of association of that data point with the corresponding meta-cluster.

4. Cluster Assignment- If the association of a data point to a meta-cluster is higher than a suitable threshold, assign the data point to the meta-cluster. Data points belonging to more than one cluster would be expected to have a high association level with all the corresponding meta-clusters and would be assigned to them.

Soft Kernel K-Means (SKK) Algorithm. Most consensus mechanisms have an intermediate step where an $n \times n$ consensus matrix M is formed from the individual clusterings of n objects. For example, the elements of M can represent the fraction of solutions in which two objects are placed in the same cluster. If one considers M as a kernel matrix, then a soft version of the kernel K-means algorithm [3] can be used to detect overlapping clusters. The analysis below assumes a mixture model of spherical Gaussians for simplicity, but can be readily extended to components from the broader family of exponential distributions.

Notation: $\phi(x_i)$ is the mapping of the data point x_i to the new space defined by the kernel. c_j represents the j^{th} cluster with parameters represented by θ_j. μ_j , σ_j^2 and $P(c_j)$ are the center, variance and prior for the j^{th} cluster respectively in the ϕ space. $h_{x_i}^{c_j}$ is the posterior probability $P(z_i = c_j|\theta, x_i)$ of the data point x_i being generated from cluster c_j.

With a kernel, the EM objective function that is maximized iteratively is $\sum_{i=1}^{n}\sum_{j=1}^{k} h_{x_i}^{c_j} \log\left(P(c_j)P(\phi(x_i)|\theta_j)\right)$. The EM algorithm requires computation of the squared Euclidean distance of each data point x_i to each cluster center μ_j, $dist(x_i, \mu_j) = \|x_i - \mu_j\|^2$. With a kernel, the center for cluster c_j will be $\mu_j = \frac{\sum_{i=1}^{n} h_{x_i}^{c_j}\phi(x_i)}{\sum_{i=1}^{n} h_{x_i}^{c_j}}$, provided the Jacobian of the kernel matrix is defined. The distance

$dist(\phi(x_i), \mu_j)$ can now be computed as:

$$\|\phi(x_i) - \mu_j\|^2 = \phi(x_i).\phi(x_i) - \frac{2\sum_{l=1}^{n} h_{x_l}^{c_j} \phi(x_i).\phi(x_l)}{\sum_{l=1}^{n} h_{x_l}^{c_j}} + \frac{\sum_{p=1}^{n}\sum_{l=1}^{n} h_{x_p}^{c_j} h_{x_l}^{c_j} \phi(x_p).\phi(x_l)}{(\sum_{l=1}^{n} h_{x_l}^{c_j})^2}$$

$\|\phi(x_i) - \mu_j\|^2$ can hence be represented in terms of dot products only. If we have a kernel matrix K, where $K_{ij} = \phi(x_i).\phi(x_j)$ then we can compute this distance without having an explicit representation for $\phi(x_i)$. The dot product terms can be replaced by the corresponding values from the kernel matrix. The pseudo-code for SKK is as follows:

1. Initialize cluster membership by assigning each data point x_i ($i = 1$ to n) to some cluster c_o such that $h_{x_i}^{c_j} = 1$ if $c_j == c_o$, $h_{x_i}^{c_j} = 0$ otherwise.

2. Compute distances of data points to cluster centers

$$dist(\phi(x_i), \mu_j) = K_{ii} - 2\frac{\sum_{l=1}^{n} h_{x_l}^{c_j} K_{il}}{\sum_{l=1}^{n} h_{x_l}^{c_j}} + \frac{\sum_{p=1}^{n}\sum_{l=1}^{n} h_{x_p}^{c_j} h_{x_l}^{c_j} K_{pl}}{(\sum_{l=1}^{n} h_{x_l}^{c_j})^2}$$

3. Compute the variance of each cluster

$$\sigma_j^2 = \frac{\sum_{i=1}^{n} h_{x_i}^{c_j} dist(\phi(x_i), \mu_j)}{\sum_{i=1}^{n} h_{x_i}^{c_j}}$$

4. Compute the prior for each cluster $P(c_j) = \frac{\sum_{i=1}^{n} h_{x_i}^{c_j}}{n}$

5. Compute the probability of each data point x_i being generated from each cluster c_j

$$h_{x_i}^{c_j} = \frac{P(\phi(x_i)|\theta_j)P(c_j)}{\sum_{j=1}^{k} P(\phi(x_i)|\theta_j)P(c_j)}$$

$$P(\phi(x_i)|\theta_j) = \frac{1}{\sqrt{(2\pi)^d \sigma_j^2}} \exp\left(\frac{-dist(\phi(x_i), \mu_j)}{2\sigma_j^2}\right)$$

6. If the change in the objective function is below a threshold stop, else loop to 2.

The resulting algorithm is very similar to the EM algorithm for a Gaussian mixture model. Steps 2, 3 and 4 correspond to the Maximization step and step 5 is the Expectation step. The algorithm hence has the convergence guarantees of EM, where the objective function is improved in every iteration and finally converges to a local optimum. For this property to hold, the consensus matrix M has to be a valid kernel matrix, i.e. it has to be positive semi-definite. We can ensure that M is positive semi definite by adding a small value to all the diagonal elements [3]. Hence K can be set to $M + \epsilon I$, where ϵ is a small value. After the SKK algorithm converges, it gives for each data point x_i, the probability $h_{x_i}^{c_j}$ of that point being generated from each mixture component c_j. If $h_{x_i}^{c_j} > \lambda$ where λ is a suitable threshold value, the data point x_i is assigned to cluster c_j. SKK could be preceded by a model selection step to allow the algorithm to find the best k when it is not specified.

Baseline Model. A simple generative baseline for comparative studies is to employ a Gaussian mixture model as the generative model and provide for each data point, the probability that the data point was generated by a particular mixture component. For allowing data points to belong to multiple clusters, a threshold value (λ) needs to be fixed. If the probability of a point being generated from a particular cluster is greater than the threshold value, the data point is assigned to that cluster. The parameter λ controls the degree of overlap. The mixture model that we consider consists of spherical Gaussians, with covariances of the form $\sigma^2 I$.

Model Selection. A model selection approach for overlapping clusters is based on the X-means algorithm [10]. A reasonable range (k_{min} to k_{max}) is input to the algorithm. The algorithm determines the best model for the data, which can then be followed by an overlapping clustering algorithm. The algorithm begins with a number of centers $k = k_{min}$ chosen randomly and continues adding centers where they are needed till k_{max} is reached. At each step of the algorithm, the threshold λ allows data points to be assigned to multiple clusters. The resulting iterative algorithm consists of a parameter improvement and a structure improvement step repeated until convergence. For further details the reader is referred to [2].

5 Experimental Results

To evaluate the overlapping clustering algorithms, precision, recall, and F-value are calculated over all pairs of points, by comparing the final clustering with the true class labels.

Datasets. The algorithms are evaluated on synthetic and real microarray data. The synthetic data used is microarray-like data with explicitly modeled overlapping clusters generated as suggested by Banerjee et al. in [1] with a clear ground truth, described in Table 1. The real dataset used is the Gasch dataset [6] consisting of the 995 most actively expressed genes of yeast under 173 environmental stress conditions. A subset of 400 genes from this dataset is used for experimentation.

Model Selection Results. Table 2 shows the results of the

Dataset	# genes	# conditions	# processes (clusters)
Small	20	10	5
Medium	75	30	10
Large	200	50	30

Table 1. Synthetic Microarray Datasets

Dataset	True k	Predicted k
Small	5	3
Medium	10	7
Large	30	21
Gasch subset	33	28

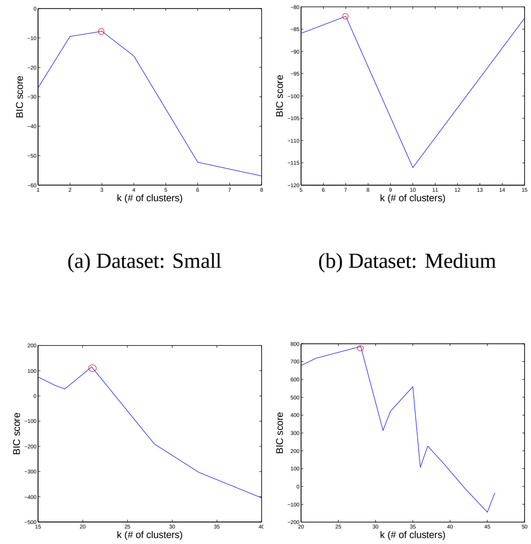
Table 2. Model Selection for Microarray Data

overlapping model selection algorithm on the synthetic and real microarray datasets, obtained over a representative run with a fixed random seed. Figure 1 shows the plot of the BIC score vs # of clusters for the microarray datasets. The value of k at which the BIC score is the highest is selected. From Table 2 we can observe that the predicted value of k is reasonably close to the true k. Another observation is that the predicted k value is consistently underestimated on these datasets. This error can be corrected by reducing the penalty term used in the computation of the BIC score [2].

Evaluation of Clustering Algorithms. Plots of the precision, recall and F-value of the clustering results of the MCLA and SKK algorithms against the threshold showed that the F-value remains relatively flat and is not very sensitive to the threshold within a particular range of values [2]. Hence, the threshold can safely be fixed at a value within this range. For the experiments in this section the threshold for all the datasets has been set to a predetermined value chosen accordingly.

Table 3 compares the results of the thresholded MCLA and the SKK algorithm to other generative approaches that detect overlapping clusters. The baseline is the Gaussian mixture model (GMM) as described in section 4. The other approaches compared to are the model based clustering algorithms DynamicM and Least Squares (Lsq) proposed by Banerjee et al. [1]. Both the Lsq and DynamicM approaches are initialized with K-means and both approaches use squared loss as the loss function. SKK is initialized using K-means. Table 3 displays in each cell the precision, recall and F-value of the clustering results, averaged over 10 runs. Figure 2 compares the F-values of the approaches described above on the synthetic and real microarray datasets.

Figure 2 indicates that the thresholded MCLA algorithm does significantly better than the simple GMM approach on all of the above datasets and consistently outperforms most of the other algorithms. SKK (k known) also does significantly better than the baseline GMM algorithm and DynamicM on all the synthetic microarray datasets. SKK (k predicted) does only slightly worse than SKK (k known) but still does better than GMM, on an average on the synthetic microarray datasets. On the medium dataset, the Lsq approach outperforms the others with very high precision and recall. However on other datasets, Lsq has very high precision but low recall and hence a lower F-value as seen in Table 3. SKK does not do as well on the Gasch dataset since its precision is comparable to the other algorithm but it has a much lower recall. The reason for the low recall of SKK on Gasch is that on this dataset the SKK algorithm converges very quickly to a mixture of very low variance, peaked Gaussian components. This causes the probabilities of data points being generated from the mixture components to be very close to either 0 or 1. The threshold does not make a difference in this case and very few overlap-



(a) Dataset: Small (b) Dataset: Medium

(c) Dataset: Large (d) Dataset: Gasch

Figure 1. BIC plots for Model Selection

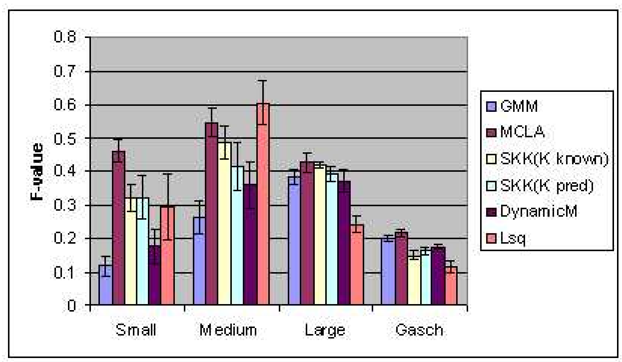


Figure 2. F-value Comparison

Dataset	GMM	MCLA	SKK (k known)	SKK (k pred.)	DynamicM	Lsq
Small	0.067 ± 0.017	$\mathbf{0.402 \pm 0.048}$	0.225 ± 0.032	0.218 ± 0.071	0.109 ± 0.034	0.252 ± 0.144
	0.507 ± 0.148	0.553 ± 0.070	0.560 ± 0.046	$\mathbf{0.800 \pm 0.251}$	0.487 ± 0.147	0.467 ± 0.104
	0.118 ± 0.029	$\mathbf{0.461 \pm 0.033}$	0.321 ± 0.040	0.321 ± 0.066	0.176 ± 0.051	0.293 ± 0.100
Medium	0.176 ± 0.041	0.521 ± 0.060	0.484 ± 0.069	0.335 ± 0.093	0.245 ± 0.053	$\mathbf{0.597 \pm 0.099}$
	0.543 ± 0.124	0.572 ± 0.028	0.492 ± 0.028	0.581 ± 0.054	$\mathbf{0.672 \pm 0.121}$	0.621 ± 0.061
	0.262 ± 0.049	0.544 ± 0.043	0.487 ± 0.049	0.416 ± 0.071	0.357 ± 0.070	$\mathbf{0.605 \pm 0.067}$
Large	0.257 ± 0.010	0.428 ± 0.131	0.275 ± 0.003	0.296 ± 0.022	0.552 ± 0.065	$\mathbf{0.731 \pm 0.051}$
	0.758 ± 0.092	0.544 ± 0.228	$\mathbf{0.876 \pm 0.046}$	0.616 ± 0.172	0.281 ± 0.040	0.145 ± 0.017
	0.383 ± 0.021	$\mathbf{0.427 \pm 0.029}$	0.419 ± 0.006	0.391 ± 0.023	0.369 ± 0.034	0.242 ± 0.024
Gasch	0.139 ± 0.004	0.133 ± 0.008	0.140 ± 0.008	0.151 ± 0.005	0.133 ± 0.002	$\mathbf{0.168 \pm 0.016}$
	0.349 ± 0.053	$\mathbf{0.634 \pm 0.240}$	0.162 ± 0.032	0.177 ± 0.021	0.254 ± 0.022	0.088 ± 0.024
	0.198 ± 0.007	$\mathbf{0.212 \pm 0.014}$	0.148 ± 0.013	0.162 ± 0.010	0.175 ± 0.006	0.113 ± 0.019

Table 3. Precision, Recall, F-value Comparison.

ping memberships are detected, leading to low recall. This problem probably occurs due to poor initialization which is currently done using a non overlapping, partitional cluster assignment with K-means.

6 Conclusions and Future Work

The experimental results demonstrate the effectiveness of the thresholded MCLA and the SKK algorithms on both, real and synthetic microarray data. Model selection for overlapping clusters consistently finds a model that is close to the true model. One of the main strengths of the developed approaches is that only the consensus matrix is required to obtain the final cluster assignments. The original dataset does not have to be stored, which is beneficial in case of very large datasets.

The SKK algorithm is based on the EM algorithm and is hence prone to poor initialization and poor local optima. Better initialization techniques that take overlapping cluster membership into account and local search techniques could improve the stability and accuracy of this algorithm. In this paper, we clustered all the data, so that the results could be compared to other partitional based approaches. For microarray analysis typically there are some genes that do not participate in any of the processes, and thus one may want to cluster only a part of the data [5]. We plan to investigate this aspect in the near future.

Acknowledgements: The research was supported by NSF grants IIS 0325116 and IIS 0307792.

References

[1] A. Banerjee, S. Basu, C. Krumpelman, J. Ghosh, and R. Mooney. Model based overlapping clustering. *Proceedings of KDD2005*, pages 100–106, 2005.

[2] M. Deodhar. Consensus clustering of microarray data. Master's thesis, University of Texas at Austin, 2006. `www.ece.utexas.edu/~deodhar/msthesis.pdf`.

[3] I. Dhillon, Y. Guan, and B. Kulis. A unified view of kernel k-means, spectral clustering and graph clustering. In *UTCS Technical Report TR-04-05*, 2005.

[4] V. Filkov and S. Skiena. Integrating microarray data by consensus clustering. In *International Journal on Artificial Intelligence Tools (IJAIT).*, pages 4:863–880, 2004.

[5] J. G. G. Gupta. Bregman bubble clustering: A robust, scalable framework for locating multiple, dense regions in data. *In Proc. ICDM: The 2006 IEEE International Conference on Data Mining*, December 2006.

[6] A. Gasch, P. Spellman, C. Kao, O. Carmel-Harel, M. Eisen, G. Stotz, D. Botstein, and P. Brown. Genomic expression program in the response of yeast cells to environmental changes. *Molecular Cell Biology*, 11:4241–4257, 2000.

[7] J. Ghosh. Scalable clustering. In N. Ye, editor, *The Handbook of Data Mining*, pages 247–277. Lawrence Erlbaum Assoc., 2003.

[8] G. Karypis and V. Kumar. Multilevel k -way hypergraph partitioning. In *Design Automation Conference*, pages 343–348, 1999.

[9] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering-a resampling-based method for class discovery and visualization of gene expression microarray data. In *Journal of Machine Learning*, pages 52: 91–118, 2003.

[10] D. Pelleg and A. Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proc. 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, San Francisco, CA, 2000.

[11] E. Segal, A. Battle, and D. Koller. Decomposing gene expression into cellular processes. In *Proc. 8th Pacific Symposium on Biocomputing (PSB), Kauaí*, Jan 2003.

[12] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining partitionings. *Proc. Conference on Artificial Intelligence (AAAI 2002)*, pages 93–98, July 2002.

[13] S. Swift, A. Tucker, V. Vinciotti, and N. Martin. Consensus clustering and functional interpretation of gene-expression data. In *Genome Biology 5:R94*, 2004.

[14] K. Tumer and J. Ghosh. Linear and order statistics combiners for pattern classification. In A. Sharkey, editor, *Combining Artificial Neural Nets*, pages 127–162. Springer-Verlag, 1999.