[Frontiers in Bioscience 13, 677-690, January 1, 2008]

Emergent unsupervised clustering paradigms with potential application to bioinformatics

David J. Miller[1], Yue Wang[2], George Kesidis[3]

[1]Dept of Electrical Engineering, Pennsylvania State University, University Park, PA 16802 [2]Department of ECE, Virginia Polytechnic Institute and State University, Arlington, VA 22203 [3]Departments of EE and CSE, Pennsylvania State University, University Park, PA 16802

TABLE OF CONTENTS

1. Abstract
2. Introduction
3. Unsupervised clustering with integrated feature and order selection
3.1. Introduction
3.2. Review of recent methods
4. Semisupervised clustering
4.1. Introduction
4.2. Review of recent methods
4.3. Formulation
4.4. Illustrative Experiment
5. Clustering in the presence of confounding variables
5.1. Introduction
5.2. Review of recent methods
6. Stability of clustering solutions
6.1. Introduction
6.2. Review of recent methods
7. Conclusions
8. References

1. ABSTRACT

In recent years, there has been a great upsurge in the application of data clustering, statistical classification, and related machine learning techniques to the field of molecular biology, in particular analysis of DNA microarray expression data. Clustering methods can be used to group co-expressed genes, shedding light on gene function and co-regulation. Alternatively, they can group samples or conditions to identify phenotypical groups, disease subgroups, or to help identify disease pathways. A rich variety of unsupervised techniques have been applied, including partitional, hierarchical, graph-based, model-based, and biclustering methods. While a number of machine learning problems and tools have found mainstream applications in bioinformatics, in this article we identify some challenging problems which, though clearly relevant to bioinformatics, have not been extensively investigated in this domain. These include i) unsupervised clustering with unsupervised feature selection, ii) semisupervised learning, iii) unsupervised learning (and supervised learning) in the presence of confounding variables, and iv) stability of clustering solutions. We review recent methods which address these problems and take the position that these methods are well-suited to addressing some common scenarios that occur in bioinformatics.

2. INTRODUCTION

In recent years, unsupervised clustering, statistical classification, feature selection, and related machine learning techniques have found an increasingly influential role in bioinformatics, as evidenced by the large number of papers involving these topics which are appearing in journals such as *Bioinformatics*, a number of recent books (37), research compendia (28), and commercialization efforts. While supervised classification plays an important role, this paper will focus primarily on unsupervised learning methods, as well as hybrid (semisupervised) techniques. Clustering methods can be used to identify co-expressed genes, shedding light on gene function and co-regulation. Alternatively, they can group samples or conditions in order to identify phenotypical groups, sub-groups of a disease, patient sub-groups that respond to drug treatment in different ways, or to segment time course data for disease pathway analysis. Some examples of clustering techniques applied to bioinformatics include e.g. (2),(66),(19). Clustering methods have also been applied to computer-aided diagnosis, e.g. (61). A variety of clustering methods have been applied in these contexts, including partitional, hierarchical, graph-based, model-based, and biclustering techniques. There are several excellent recent surveys of clustering methods (64), clustering applied to gene expression data (28), and

biclustering methods (36). This paper aims to be complementary to these articles, covering several machine learning issues with, we argue, high relevance to bioinformatics, and yet which have not been extensively addressed in past studies. These include i) unsupervised clustering with unsupervised feature selection, ii) semisupervised learning, iii) unsupervised learning (and supervised learning) in the presence of confounding variables, and iv) stability of clustering solutions. We next identify these problems, review existing work, and in some cases propose new approaches.

3. UNSUPERVISED CLUSTERING WITH INTEGRATED FEATURE AND ORDER SELECTION

3.1. Introduction

Microarray expression data sets consist of the simultaneous measurement of expression levels for thousands (as many as tens of thousands) of genes, for each tissue sample in an experimental study. The samples may come from different patients. Alternatively, they may come from the same patient but under different experimental conditions. Consider the objective of clustering samples (or conditions). In a typical study, there may be less than a *hundred* samples. Thus, this problem amounts to clustering a *very* sparse data sample, within a *very* high-dimensional space. This is a nontrivial problem even if both the number of groups in the data (e.g. (disease present, disease absent)) *and* the relevant gene feature subspace (consisting of the genes that are most characteristic of particular groups and those most discriminating between groups) are known. Even given this information, there is the difficult choice of clustering dissimilarity measure (equivalently, the choice of parametric statistical form for a mixture model/model-based clustering solution (3)) and the challenging nature of clustering as an optimization problem (27), with sensitivity to parameter initialization for local optimization methods and high complexity for global optimization methods. These aspects are hurdles to achieving accurate, effective solutions. However, in the most general unsupervised setting (and in many practical bioinformatics contexts), the number of clusters *and* the relevant gene subspace are both *unknown* and need to be estimated in an unsupervised fashion, jointly with the (accurate) partitioning of the data samples into groups. This is an extremely challenging version of the clustering problem, and yet one which has been assailed by several recent methods, as will be reviewed in the next section.

3.2. Review of recent methods

Most approaches to unsupervised clustering and feature selection perform some type of alternating optimizations, with clustering performed given selected features and then, alternately, with feature selection treated as a *supervised* problem, to maximize some measure of discrimination between the (current) clusters treated as classes. The general notion behind these methods is that removing noisy features should improve clustering accuracy and cluster separation which, in turn, should make it easier to find "clean" features that well-discriminate these clusters. Thus, both clustering and estimated gene space accuracy should tend to improve with the successive optimizations.

Some early approaches were in fact developed specifically for microarray data. In (62), the authors first chose an initial set of genes based on their individual power to discriminate components in a two-component Gaussian mixture model for the data. They then alternately performed graph-based clustering and feature filtering steps, with the latter based on a supervised selection criterion and on a Markov model, with (redundant) genes rejected if they fall in the "Markov blanket" of other genes. In (54), the authors first clustered in the gene dimension, forming k gene subspaces from the gene clusters. For each subspace, they then partitioned the samples into two clusters. They then defined "cross-product" groups, *i.e.* sets of samples which all fall in cluster i in gene space 1 and cluster j in gene space 2 $\forall i, j$. There are 2^k such sample groups, denoted $C_1, C_2 ..., C_{2^k}$, mutually exclusive and collectively exhaustive of the samples. They then further pooled selective pairs of cross-product groups to form "heterogeneous" groups – these are pairs of cross-product groups whose cluster labels are different from each other, for every gene subspace. Thus, a heterogeneous group is a set of samples that is well-discriminated into two distinct groups, for each gene subspace. Heterogeneous groups were used to guide gene selection – each gene's vector across samples was correlated with a representative vector from a heterogeneous group. Essentially, the genes most highly correlated with the group were retained and the remainder discarded. This sequence of steps represents one iteration of the method. The next iteration begins again with gene clustering (starting from the now reduced set of genes). (54) only addressed clustering samples into 2 groups.

There are several disadvantages to the aforementioned methods. First, neither method solves the sample clustering and feature selection tasks in a way that is consistent with minimization of a common objective function. Thus, there is no mathematically well-defined sense of convergence for these methods and also no reference objective function for assessing solution quality (except for comparison to ground-truth biological knowledge (if available) on the groups and relevant genes). Second, these methods do not estimate the number of clusters in the data – this must be known *a priori* or set by the user. Third, both methods require the (user-subjective) choice of threshold parameter values. These values will certainly affect the results – e.g., (54) uses a threshold to control the number of retained genes. Finally, both methods are greedy in the sense that each iteration further *reduces* the number of genes, with no ability to "resurrect" a rejected gene in subsequent iterations, even if the current clustering would warrant this gene's reinclusion. Recently, several new methods have been proposed from within the machine learning community which address some of the aforementioned shortcomings. These methods are next described.

There are three basic strategies in the literature for combined clustering and feature selection. The simplest

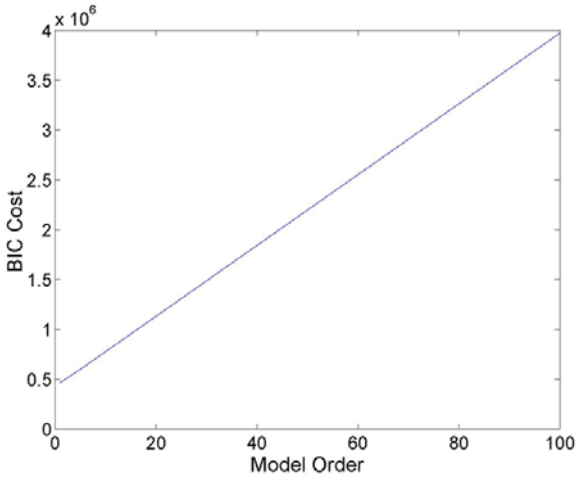

Figure 1. BIC curve for a naive Bayes mixture applied to *Reuters* text documents. Note that the predicted (minimum BIC) order is one component, a grossly inaccurate estimate for this data set with 22 topics.

(but in principle the least accurate) is to first perform front-end dimensionality reduction and then cluster in a now (vastly) reduced feature space. For example, in the text domain, (35) used a singular value decomposition (SVD) to first reduce from thousands down to 20 features. However, the choice of 20 features was author-selected and appears somewhat arbitrary. Moreover, principal component analysis (PCA) and related approaches are optimal with respect to minimum mean-squared error approximation of the original data, but may not best preserve group discriminability. Clearly, it should be better to choose features given knowledge of the clusters being sought, rather than "blind" feature selection. Finally, use of PCA/SVD for dimensionality reduction maps the data to a new space. Thus, some interpretability in the original space is lost – in our case, information on which genes are truly relevant to a given cluster may be obscured by this process.

A second strategy is *wrapper-based*, wherein one generates numerous clustering solutions, for different candidate feature subspaces (which may be chosen either randomly or through some directed search), with all solutions compared with respect to a common clustering fitness function. One such approach is (18), which used greedy forward search for feature selection, EM for mixture learning, and a clever "cross-projection" criterion that allows "level playing field" comparison between clustering solutions defined on different feature spaces. The wrapper-based advantage over the previously mentioned methods is that now there *is* a common criterion for assessing the clustering solutions (and also, thus, for guiding the search for solutions). However, the set of possible feature spaces is vast – if each cluster uses the same feature space, there are 2^D possible spaces, with D the number of dimensions (genes). Greedy forward or backward feature selection methods will search a very small portion of this set and are likely to find quite suboptimal solutions. Random selection likewise will require generation of a vast number of candidates, with clustering needing to be performed and fitness evaluated for each one, to find good solutions. This may entail vast (and impractical) computational complexity. Moreover, the choice of the number of clusters further greatly expands the search space (and requires model order selection criteria that are effective for high-dimensional data, as next discussed).

A final approach is developed from the viewpoint that *feature* selection is simply another facet of model selection, along with choosing the number of components (model order). Order selection is often performed using statistical penalty functions such as minimum description length/Bayesian Information Criterion (BIC) (45). BIC is defined as follows:

$$BIC(M) = \frac{|\Theta(M)|}{2}\log N - \log P[\chi \mid \Theta(M)], \qquad (1)$$

where $X = \{\underline{x}_i, i=1, \ldots, N\}$ is the data set, N the number of samples, M the model order, $\Theta(M)$ the parameter set at this order (whose values specify a parametric statistical model), and $|\cdot|$ the number of free parameters in the set. The first term in BIC penalizes complex models while the second (log-likelihood) indicates how well the model explains the data. In (22),(23) the question was raised of whether one can jointly optimize *all* parts of the solution – the selected features, the number of components, and the sample partitioning – with respect to a single objective function, this model penalty function. Criteria such as BIC, which involve a data fitness term and a model complexity term, seek the order that can be supported by the given (finite) amount of data. Why should the same principle not also be applicable to the choice of features? There are several recent works which follow a related strategy (5),(34),(23). The first two, however, were not intended for high-dimensional data – (5) only considered $D = 10$ and (34) only tried D up to 47. We will focus on (23), which was motivated particularly by the high-dimensional text document domain and which developed a suitable model.

(23) considered "naive Bayes" mixtures (diagonal covariances in the Gaussian case). In such models, there are k parameters per dimension ($k = 2$ in the Gaussian case) for each component in the model. In a text document database with 2000 articles and ~ 10,000 data (word) dimensions (and 20 ground-truth components/topics), this amounts to more than 200,000 parameters for only 2000 data points[1]. Standard application of BIC in this case will *grossly* underestimate the model order because the model complexity cost of each additional component is too high, relative to its benefit to the log-likelihood. An example from (23) is shown in Figure 1 where the BIC-estimated order is one (component), even though there are 22 ground-truth topics. The situation is even worse for microarray data, where the dimensionality is similar but the number of data points is at least an order of magnitude smaller in most studies. As explained in (23), the fundamental problem here is not the criterion, BIC. It is the fact that there are insufficient degrees of freedom in the naive Bayes mixture for trading data fitness for reduced complexity. (23) proposed structured naive Bayes mixtures that allow

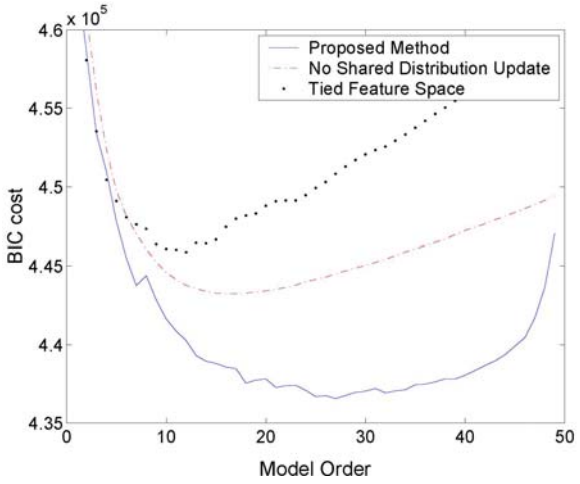

Figure 2. BIC curves for several variants of naive Bayes mixtures applied to *Reuters* text documents. The best approach, from (3), gives each component flexibility in the choice of its feature space and optimizes all model parameters. This method estimates a model order of 25 components for this 22 topic data set.

sharing of parameters across components, *i.e.* the likelihood model:

$$P[\underline{x} \mid \Theta(M)] = \sum_{j=1}^{M} \alpha_j \prod_{k=1}^{D} P[x_k \mid \theta_j]^{v_{jk}} P[x_k \mid \theta_s]^{(1-v_{jk})} \quad (2)$$

Here, $P[x_k|\theta_j]$and $P[x_k|\theta_s]$ are component-specific and shared distributions, respectively, with $v_{jk} \in \{0, 1\}$a *binary switch variable*, where

$$V_{jk} = \begin{cases} 0 \text{ if component j uses a shared model for } x_k \\ 1 \text{ if component j uses a tailored model for } x_k \end{cases}$$

These variables specify the informative features for each component. We emphasize these variables are *model parameters* that need to be learned. In the text domain, these variables determine topic-specific keywords and for microarrays they determine the "informative" genes for each cluster, *j*. This type of "parsimonious" mixture was previously introduced in (24). A crucial aspect is an efficient method for coding the model parameters, which was developed and described in (23).

In (23) it was proposed to directly minimize BIC in determining *all* parts of this model – the order M, the component parameters, the switching variables, and the data partitioning. All of this was done in a purely unsupervised fashion. There are several benefits to this model. First, it offers large flexibility in trading off model fitness for complexity – the $\{v_{jk}\}$ switches are additional parameters, but they allow sharing distribution parameters across multiple components, which can greatly reduce the number of model parameters and thus the model complexity. If a component does not use a shared model for a given feature, we say the feature is "informative" for the given component. As the number of "informative" features are reduced, more components can be supported by the limited data and, thus, accurate model order selection can be achieved using BIC, even on huge feature spaces. An example from (23) is shown in Figure 2 where optimizing BIC for various model orders yields an optimal model at order 25 for a document data set with 22 ground-truth topics. Order 25 is also best in a generalization (test set likelihood) sense. Even though D =10,000 in this case, the total number of "informative" features is only ~ 500. Thus, choosing the model to minimize BIC yields a sparse set of informative features. This is suggestive, for microarrays, of finding a sparse set of informative genes associated with a disease group. In addition to facilitating accurate cluster number estimation (and accurate data partitioning (23)) in high dimensions, the model in (23) is also interesting in that each component/cluster has its own set of informative features, i.e. its own feature space. Most methods tie the feature space across all the clusters. However, for example, individual genes may only be relevant to a subset of disease groups/clusters. This is captured by the model in (23). In Figure 2, it can be seen that tying the feature space across components performs poorly, underestimating the number of topics in the data. Allowing a customized "informative feature space" for each component is more efficient in allocating the model complexity across components and gives much better results. This flexibility in defining the feature space for each component is related to the representation capability of biclustering – note that biclusters are defined by subsets of genes and subsets of conditions (36). The mixture model in (23) likewise captures a subset of conditions/samples within a cluster that is defined over a customized informative gene subset. Unlike many biclustering methods, though, (23) postulates a stochastic generation model for the data array. Moreover, based on optimization of BIC, this method gives a statistically principled approach (and accurate approach for the text domain (23)) for estimating the number of clusters.

The mixtures in (23) are learned via a generalized EM algorithm, embedded within a model order reduction procedure, which *directly* minimizes BIC at each order. Thus, the solution is locally optimal with respect to BIC. At each order, within the optimization, both switch values are evaluated (multiple times) for each gene – thus, "uninformative" genes can switch to "informative", or vice versa, for a given cluster (unlike the greedy methods, which solely shrink the "current" feature space). This method was tested and demonstrated favorably against several other unsupervised techniques such as (34),(56),(51) on UC Irvine Machine Learning data sets and on text. We believe this approach should be useful for microarray data. One limitation, however, is the naive Bayes (component-conditional feature independence) assumption. Given a solution for the Gaussian case and its associated data partition, one can estimate (nondiagonal) covariances and change the model to include these correlation terms. However, this new solution will increase model complexity and could increase the BIC cost.

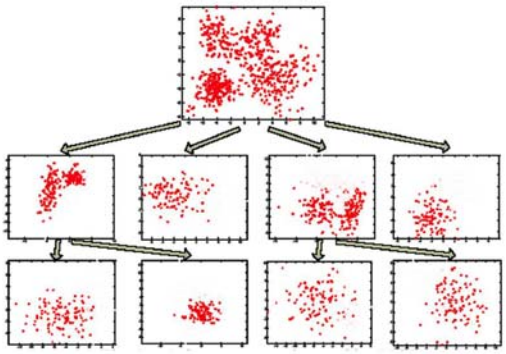

Figure 3. An example of coarse-to-fine data partitioning obtained using VISDA.

4. SEMISUPERVISED CLUSTERING

4.1. Introduction

In some cases, the data set may contain some *partial* class information which, while insufficient to allow the learning problem to be treated as supervised classification, can still help to "guide" clustering solutions, so as to be most relevant to the ground truth classes present in the data. One possibility is that the class of origin may be known for a labeled subset of the training set, *e.g.* (46),(40). Parameter estimation based on unlabeled data, in addition to labeled samples, can in some cases improve the accuracy of class-conditional models that will be used in a pseudo-Bayes classification rule. However, there is a cautionary tale here (14). There are also purely "discriminative" learning methods for building classifiers while trying to make use of mixed labeled/unlabeled data e.g. (9),(11). Another point of view developed in (38) is that the labeled data can be effectively used to "label" the learned clusters, identifying the subset of clusters that contain known content (i.e., clusters which own at least some labeled data, in addition to possibly owning unlabeled samples). Clusters that do not contain *any* labeled samples, by contrast, may contain *novel* content, *i.e.* heretofore "unknown" classes. A semisupervised class discovery procedure was thus defined, which identifies putative unknown classes (purely unlabeled clusters in the data) relative to the existing, known classes in the data (38). In principle, this approach could be used to identify new subtypes of a disease – in a semisupervised patient sample where all patients are known to be sick, but with only a subset labeled by accurate disease diagnosis, learned compact clusters of samples that are strictly unlabeled may be taken as putative new disease subtypes (with this tentative hypothesis tested through further analysis). While there has been some investigation of related ideas on biological data (50), semisupervised class discovery has not been substantially investigated in the biological domain.

Another case of significant interest in biosciences is wherein there is partial supervision in the form of user interaction to guide the search for grouping structure in the data. One such approach, known as *visual and statistical data analyzer* (VISDA), was developed in (59). This is a top-down, hierarchical, soft (mixture-based) clustering scheme that, at each level of the hierarchy, linearly maps the (high-dimensional) clusters at the current level to a 2-D subspace. This allows users both to visualize the current clusters and to assist accurate estimation of the number of sub-groups (and their initial centroids) for clustering at the next level of the hierarchy. The EM algorithm is applied to optimize the clustering at each level, with the minimum description length criterion, along with user interaction, used to estimate the number of clusters. Both the hierarchical nature of this clustering and the user interaction, which allows coarse-to-fine structure exploration, have obvious appeal for bioscience applications. While user interaction is one form of "partial supervision", VISDA also has a semisupervised mode wherein there is label knowledge for some samples. The labels are represented by color-coding the samples in the 2-D visualization space. This can further assist accurate sub-clustering at the next level. VISDA has recently been adopted as a standard data analysis tool by the National Cancer Institute, as part of the caBIG initiative (60). An example of (visualizable) top-down clustering of high-dimensional data obtained by VISDA is shown in Figure 3. An application of VISDA to create a pathologically plausible hierarchy of thirteen distinct muscular dystrophy phenotypes was reported in (68). The learned class hierarchy is shown in Figure 4.

Also relevant to molecular biology is the case where supervising class labels are not available but where, instead, there may be a set of *must-link* (ML) and *cannot-link* (CL) constraints, each indicating a pair of samples that, respectively, should or should not be assigned to the same group (58). For example, for gene clustering, it may be known that a particular pair of genes is involved in the same biological function (and hence should belong to the same cluster or class). It may likewise be known that certain pairs of genes should *not* belong to the same cluster. For clustering expression profile samples, it may be known, e.g. using supplementary measurement modalities other than gene expression such as radiological images or DNA sequence motifs, that a pair of samples should (should not) belong to the same group. Some related work applied to gene expression data is (41), where knowledge of a common function for a subset of the genes was incorporated, for gene clustering, via a tied mixture model, wherein all genes in the subset use the same mixture priors in associating to the mixture components (clusters) in the model. This approach represents a particular method for "soft" imposition of constraint knowledge while peforming clustering and it imposes "must-link" information only. More generally, one can incorporate both must-links and cannot-links. In general, these constraints can help to overcome cluster initialization sensitivity, suboptimality in the choice of the clustering distortion metric (e.g., they may be used to *learn* this metric (63)), and as will be discussed, they can even help to estimate the number of classes in the data. Several such methods are next reviewed in more detail.

4.2. Review of recent methods

Constraint information could be elicited from domain experts via on-line interactive databases, where (multiple) users/experts may specify must-links and cannot-

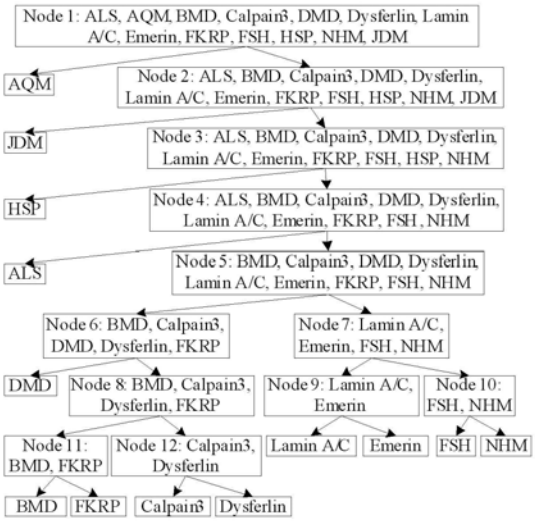

Figure 4. A pathologically plausible tree hierarchy for 13 muscular dystrophy phenotypes obtained using VISDA.

links for given (or user-selected) pairs of examples. In this setting, it may be more appropriate to specify must-link and cannot-link information, rather than class labels, because users (even experts on a given domain) may not agree on the number of classes, class names, or even defining class attributes. Constraints can be solicited without even explicitly agreeing on class definitions or on the number of classes. A number of prior works address clustering with constraints. (58) develops a variant of K-means that enforces the learned clusters to be consistent with the given constraints. (49) introduced constraints within graph-based clustering applied to image segmentation. (31) developed an approach suitable for hierarchical clustering. (48) incorporates hard constraints within mixture model-based clustering. In recent work (67), a new approach was proposed for learning Gaussian mixtures while agreeing with ML and CL constraints. This approach differs from prior works in several respects. First, prior works do not make a distinction between *clusters* and *classes*. In these works, e.g. (58),(48), the individual clusters are treated as distinct classes, to be learned consistent with the specified ML and CL constraints. However, some individual classes may not be accurately modeled by a single cluster – they may require multiple clusters, i.e. a mixture, for their accurate representation. An illustrative example is shown in Figure 5. Whether or not this is the case depends on the feature space, the chosen distortion metric, and on the class definitions, i.e. whether they are very narrowly or broadly defined. In the latter case, we would expect multiple clusters to be helpful for modeling some classes. Learning the metric (4),(63) may mitigate model bias associated with assuming one cluster per class. However, multiple clusters will still afford greater flexibility. Second, most prior works assume the number of clusters (and hence the number of classes) is *known*. In (67), neither the number of clusters nor the number of classes need to be assumed known – the cluster number is estimated via a model selection criterion (BIC), with the class cardinality first upper-bounded by the estimated number of clusters, and then estimated as a byproduct of the learning, so as to satisfy the constraints.

4.3. Formulation

Suppose there are K clusters in the data, belonging to (at most) $L_{\max}$ classes, with $K \geq L_{\max}$. Let the matrix $[C_{ij}]$ specify the constraint information, where $C_{ij} \in \{-1,1,0\}$ indicating, respectively, that samples i and j are must-linked, cannot-linked, or without a constraint. We can then compose a clustering cost function (complete, penalized negative data log-likelihood) that embodies both fitting the data and satisfying the constraints (67):

$$U(M,V,\Theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} M_{ik}V_{kl}\log\left[\alpha_k \beta_{l|k} f(x_i \mid \theta_k)\right] + \frac{P_0}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}\sum_{l=1}^{L}\left(\sum_{k=1}^{K} M_{ik}V_{kl}\right)\left(\sum_{k'=1}^{K} M_{jk'}V_{k'l}\right), \quad (3)$$

where $M_{ik} \in \{0,1\}$ indicates whether sample i belongs to cluster/component k, $V_{kl} \in 0,1\}$ indicates whether or not component k (M_k) belongs to class l (C_l), α_k is component k's mass, $f(\cdot|\theta_k)$ is k's density, based on its parameter set θ_k, and where $P_0 > 0$ is a constraint violation penalty. The probability mass function (PMF) $\{\beta_{l|k}\}$ gives the probability a given component belongs to each of the classes – these PMFs start out uniformly distributed but the effect of the penalty term is to drive them to {0, 1} values. Note that these PMFs allow more than one component per class, i.e. whenever $\beta_{l|k} = \beta_{l|k'} = 1$, $k \neq k'$. Also, if $\beta_{l'|k} = 0 \; \forall \; k$, then class l' is not used. $L_{\max}$ minus the number of unused classes gives the estimated class number. (67) identifies conditions under which the constraint information is sufficient for uniquely specifying the number of classes in the data. An Expectation-Maximization (EM) learning algorithm (16) that builds in a mean-field approximation is used to optimize this penalized log-likelihood in learning the mixture solution.

4.4. Illustrative Experiment

Figures 5a-c give an illustrative example. The data set is 2-D, consisting of 3 isotropic Gaussian components (with assumed known variance) and 2 classes, with one class owning 2 components; 15% of the points come with (ML or CL) constraints. The method in (48) assumes one mixture component/cluster per class and requires specifying the number of classes. This method learns general covariances for individual components, so that the cluster shape can be adapted to better satisfy the given constraints. If this method assumes 2 classes (Figure 5a), it has difficulty capturing 2 ground truth components within one of its learned classes/clusters. On the other hand, if 3 classes are assumed (Figure 5b), ML constraints within one of the ground truth classes (between 2 ground truth components) make it difficult to capture the true cluster structure. In Figure 5c, we show the result of (67), which allows multiple components per class. The proper number of mixture components (three) is accurately estimated via BIC applied to a mixture first learned without using the constraint information. We then allowed a maximum number of 3 classes and applied the EM algorithm with mean-field approximation to minimize the

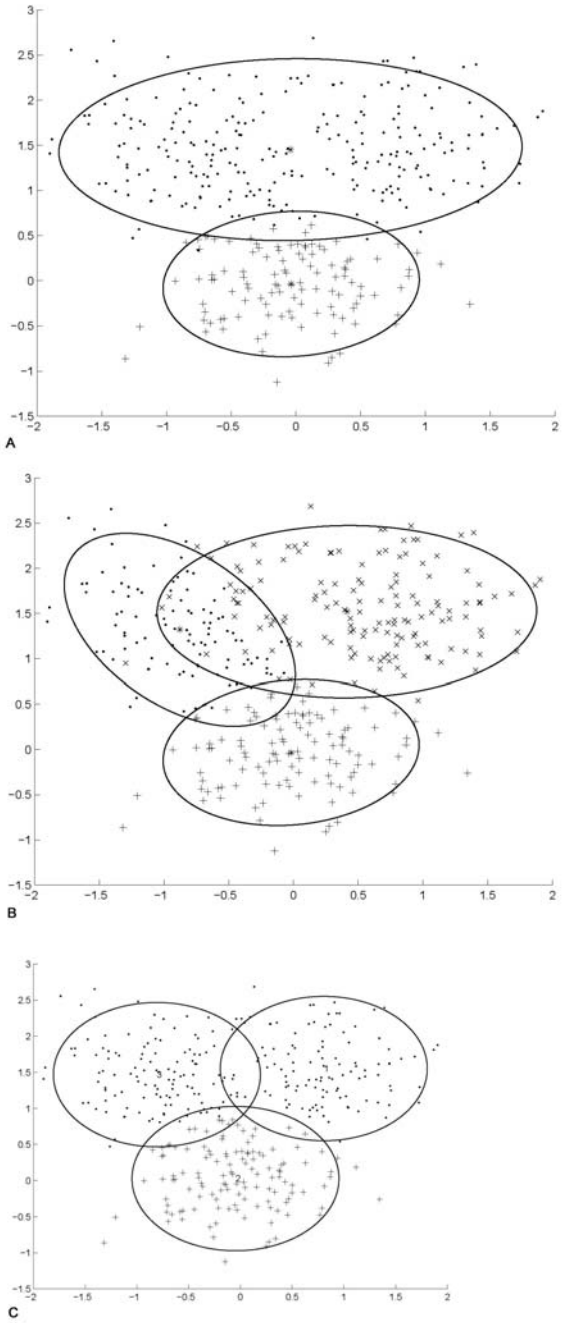

Figure 5. Mixture model solutions (given constraint information) for (48) assuming 2 classes (a)) and 3 classes (b)) and for the approach from (67) (c)), which directly estimates 2 classes in the data.

penalized likelihood (3). The optimization chose $\beta_{3|k}$=0, k= 1,2,3. Thus, the method accurately estimated the true number of classes as two. The method also gave superior mixture model fitting, as seen from Figure 5c. More comprehensive experimental results are given in (67). There, a semisupervised class discovery approach, analogous to (38), was also proposed for the case where supervision consists of constraints – in this case, a learned cluster that does not own any samples that possess constraint information is treated as a putative new class.

5. CLUSTERING IN THE PRESENCE OF CONFOUNDING VARIABLES

5.1. Introduction

A fundamental assumption in much of the clustering literature is that there is a *single* source of clustering/grouping tendency in the data – e.g., in a medical patient study, in applying a clustering method one would expect to discern groups representative of disease categorization: disease presence, disease absence, and perhaps disease subtypes. Likewise, in clustering a text document database, one would expect to reveal the underlying topics. However, there are other possibilities. One is that there is *no* clustering tendency. Another is that there are *multiple* sources of clustering tendency for a given data set. In the case of text documents, one can group by topic or by author (or by writing style). Several methods have recently been proposed for successively generating multiple clusterings, exhibiting nonredundant group structures, from a given data set. The main perspective in these papers is that there are multiple informative sources of group tendency and that the desired structure depends on the user's application/interest. However, another point of view which, we argue, has great relevance for bioinformatics is that some sources of group structure are attributable to *confounding variables*, variables of no interest for the given application but which do have influence on measured variables. These variables may be an irrelevant/nuisance source of group structure, as well as a source of measurement variability/noise.

As one example, in a microarray study involving e.g. leukemia, the proportion of patients who are smokers may be unusually high. Smoking may have strong influence on measured gene responses. Thus, unsupervised clustering may reveal smoking/nonsmoking, rather than the disease groups. As another example, multiple institutions frequently conduct studies on the same diseases, under similar experimental protocols (same treatment course, same measured variables). The number of patients in a single study is generally quite small (from tens to several hundred) due to high cost/subject availability issues, which severely limits statistical confidence in hypotheses made on the basis of learned models. One is thus tempted to pool data from multiple studies, in order to increase the statistical power of the sample. However, even given *identical* experimental protocols, there are often systematic differences in equipment and in sample processing/measurement whose effects on measured variables may *dwarf* those stemming from presence/absence of disease. Thus, clustering such pooled data may simply split the data by institution, rather than unearthing the disease structure. Several recent papers have developed methods which seek to *account* for confounding effects while clustering data. We will review some of these methods shortly and also suggest some alternative approaches.

While confounding variables present a severe challenge to unsupervised clustering, wherein the predicted

(group) variable of interest is unknown and needs to be "discovered", they also present a barrier to knowledge extraction in a supervised classification context, despite the fact that the predicted variable is in this case known. In bioinformatics, it is often just as important to identify biomarkers for disease as it is to build models that accurately predict disease presence in a patient. Supervised feature selection techniques, e.g. (32) are often used to identify the biomarkers as a small subset of the full gene space (25). However, confounding variables can "deceive" feature selection algorithms. For example, consider a patient study where, coincidentally, a high proportion of patients with leukemia also happen to be smokers [2]. If one simply ignores this possible confounding influence, the selected biomarkers may be more indicative of the confounder than of the disease. Clearly, if the confounding variable is measured, feature selection should try to account for/correct for its effects in the data. We will discuss several strategies for achieving this in section 5.3.

5.2. Review of recent methods

Several works in bioinformatics have addressed how to best combine multiple small batches of microarray samples collected by different institutions, on different platforms, or on the same platform but at different times or using different protocols. In principle, pooling sample batches to increase the number of data samples can help to achieve improved accuracy of subsequent unsupervised clustering or (supervised) statistical classifier design. Likewise, improved statistical power can be achieved if the goal, rather than data clustering, is to make inferences on the individual genes or gene subgroups involved in a disease process. However, the institution, platform, or protocol are confounding variables which may introduce systematic biases in microarray measurements. Several different approaches have been proposed to "correct" systematic biases prior to pooling the multiple microarray sample batches (1),(8),(29). These methods assume that the different batches represent the *same* underlying data groups – either a single population (*e.g.* a "disease" group) or a mixture of populations (*e.g.*, a "disease" group and a "control" group). The premise is thus that marked statistical differences exhibited by the different sample batches are primarily attributable to the systematic differences in the measurement environments associated with each of the sample batches. The methods in (1),(8), and (29) modify the samples in each data batch so as to "correct" these systematic biases. The method in (1) is based on a singular value decomposition (SVD) of the pooled data. The premise is that the variation along the principal direction(s) in the pooled data is primarily a result of systematic bias. Thus, the microarray batches should be separately altered such that, once pooled, the variation along the principal direction(s) is greatly decreased (or even wholly removed). In (8), the authors point out that this SVD-based approach will fail if the variation due to systematic bias is only comparable to or smaller than that due to group differences (between "disease" and "control") – in this case, "correcting" the sample batches in the principal direction(s) will in fact remove information that is needed to distinguish the different data groups.

A key limitation of the method based on SVD is that it does not exploit the batch index of origin, known for each sample in the pooled data set. (8) capitalized on this information by essentially treating the problem as one of statistical *classification*, with each data batch representing a different class. The direction in the data that allows best discrimination of the sample batches from each other is (reasonably) assumed to be the direction along which variation is primarily due to systematic bias. The authors in (8) thus proposed to correct the component in each sample batch that lies along this "discriminating" direction. The corrected batches will then *overlap* along this direction and no longer be discriminable from each other. The method in (8) was dubbed *distance-weighted discrimination* (DWD). There are two further aspects of DWD. One is that the "correction" amounts to removing the sample mean of each batch, projected along the "discriminating" direction. Effectively what is being assumed here is that the systematic bias only amounts to differences between batch means. The variance along this direction which remains following batch correction is thus expected to be "genuine" variation in gene responses across the pooled sample population. Second, the authors proposed a special criterion for choosing the "discriminating" direction. One possibility is to apply the linear support vector machine (SVM) method (15), *i.e.* learn a hyperplane classifier that maximizes margin (minimum distance) to the decision boundary and then choose the "discriminating" direction as the normal to the learned hyperplane. However, in (8) the authors argue that correcting in this direction introduces artificial statistical character in the data – in particular, "bunching up" of samples at the margin distance to the hyperplane. Instead of using the SVM solution, the authors proposed to choose the hyperplane to maximize the sum, over *all* samples, of inverse distances to the hyperplane. (8) demonstrates on real microarray data sets that their method achieves good "mixing" of data batches from two different microarray platforms and substantially better mixing than that achieved by the SVD method. The authors state that their approach works best when there are at least 25-30 samples per batch.

In (29), the authors sought to develop a method that corrects systematic effects when the batch size is even smaller, while achieving robustness to outlier samples. Toward this end, they proposed an empirical Bayes (EB) method that explicitly models systematic effects via batch-specific means and variances for each gene. Their approach *standardizes* gene measurements across the batches by correcting based on mean and variance estimates for each batch. Improved parameter estimation (for the small sample case) and outlier robustness are achieved by "borrowing strength" across genes, based on empirical Bayes estimators that assume priors on parameters that are identically distributed across genes for a given batch (and, thus, with hyperparameters estimated using measurements for all genes in the batch). The authors demonstrated their approach on four small microarray batches obtained from the same platform but at different times. Each batch consisted of the same two groups – a "control" group and "treatment" group. They showed that if one simply pools the four batches, subsequent hierarchical clustering groups

samples based on their *batch* of origin, rather than based on "control"/"treatment". By contrast, clustering data standardized by the EB method yielded the proper cluster structure. Moreover, EB was shown to be more robust to outliers than a simpler standardization procedure that does not perform Bayesian estimation. Finally, an advantage of EB over DWD is that EB is naturally suited to pooling two or *more* batches, while DWD, based on the solution of a binary classification problem, is naturally suited only to pooling pairs of batches.

One concern with all of the above methods is that they each perform some type of irrevocable modification of the measurements in the batches as a precursor to batch pooling (and subsequent clustering, classification, or other data analysis). Such "correction" is *inherently* a source of information loss (10). Equivalently, these procedures will introduce statistical artifacts in the data if the underlying assumptions about systematic effects (*e.g.* that they alter means and variances of gene expression) are incorrect. We next discuss several machine learning approaches that address confounding effects specifically in clustering *without* requiring any modification of the original data measurements and which, again unlike the previous methods, do not make parametric modeling assumptions about systematic effects.

Several machine learning works that seek clustering solutions nonredundant with certain known (but irrelevant) structure in the data were developed as extensions of the *information bottleneck* (IB) algorithm (56). The objective of IB is to compress one random variable X while preserving as much information as possible about a related random variable, Y (or a collection $\underline{Y} = (Y_1, Y_2, \ldots, Y_D)$). In the context of clustering DNA microarray samples, $X \in \{1,2,\ldots,T\}$ could represent the patient (sample) index, with $Y_i, i=1,\ldots,D$ the response of gene i. In this case, the index set is being "compressed" (partitioned) into subsets. While there are IB formulations that work with continuous-valued random variables, in general this requires the choice of a parametric density form. The basic IB approach works with discrete-valued random variables and thus avoids this issue. Thus, in the case of continuous-valued gene expression, some discretization (e.g., quantization) may be needed to apply IB. The IB approach creates a random variable on the cluster index set $C \in \{1,2,\ldots, M\}$, $M < T$, and views this as a "compressed" version of X. IB chooses probability mass functions (*i.e.*, a soft partition) $\{\{P[C=c|X = x], c = 1,\ldots,M\}, x = 1,\ldots,T\}$ to preserve maximum mutual information $I(C;\underline{Y})$ with Y while compressing X as much as possible, *i.e.*, minimizing $I(C;X)$. That is, IB poses and solves the constrained problem: max $I(C;Y)$ subject to $I(C;X) \leq I_0$, with this optimization solved with respect to the soft data partition.

The IB framework, based on mutual information, is a convenient one for seeking clustering solutions that are nonredundant with some known group structure in the data. Suppose we have a random variable K which represents a confounding influence/known grouping of the data, e.g. $K \in$ {"smoking", "non-smoking"}. (13) proposed to avoid redundant clusterings by penalizing the information the learned clustering possesses about K. Specifically, they posed the new problem: max $I(C;\underline{Y})-\gamma I(C;K)$, subject to $I(C;X) \leq I_0$, again optimizing with respect to the soft data partition. They demonstrated the efficacy of this approach in a document clustering context.

(20) argued that in practice it may be difficult to choose a proper value for γ. They proposed an alternative IB extension which accounts for the known information K in another natural way: by *conditioning* on it rather than by penalizing solutions that contain information about K. Specifically, (20) first proposed the problem: max $I(C;\underline{Y}|K)$, subject to $I(C;X) \leq I_0$. A potential difficulty with this problem, as noted in (20), is that this objective is invariant to arbitrary permutations on the cluster index set $\{1,2,\ldots, M\}$ given a particular value $K = k$, *i.e.*, the solution lacks global coordination across different conditioning values, $K = k$. (20) addresses this by imposing the additional constraint $I(C;X) \leq I_{\min}$, which favors solutions with global coordination of cluster labeling (*i.e.*, same meaning for cluster label values for each value $K = k$). (20) demonstrated results on clustering face images by gender and on document clustering. The authors in (20) also developed an alternative nonredundant scheme based on *ensemble clustering*, *e.g.*, (57),(53). Here, they first partitioned the data into L groups, with group l consisting of the samples with confounding value $K = l$. They then performed clustering within each group, yielding L different (local) clustering solutions, each with J clusters. Each of these (local) solutions was then *extended* to define a partition of the entire data set. Thus, at this stage, there are L different partitions of the whole data set, each with J clusters. Finally, they applied ensemble clustering techniques to form a consensus clustering from the L different partitions. The key idea here is that the initial division of the data set by confounding value removes the influence of the confounding/redundant variable.

While the above described methods do represent advances for a very challenging learning problem, there are some limitations to these approaches. By conditioning on K, the method in (20) requires estimation of the "third order" probabilities $P[Y_i|C = c, K = k]$. For bioscience applications with limited data samples there may be insufficient data to accurately estimate these probabilities. Consider in particular the case where data is pooled from multiple institutions in order to increase statistical power. Conditioning as in (20) effectively undoes this data pooling. Similarly, the method in (21) divides the data into separate sets based on the confounding value $K = k$ and separately clusters each such set. However, if the data with some value $K = k$ is very limited, this data set may be insufficient for learning a reasonable clustering solution and the extension of this (local) solution to a partition for the entire data set may be unreliable. Also, some clustering algorithm solutions cannot be easily extended to give partitions on new data (in this case, the whole data set), in particular agglomerative hierarchical clusterings.

Another limitation of the previous approaches concerns application to microarray data. As dicussed

earlier, one is often not only interested in the underlying phenotypical groups but also in the dominant genes (biomarkers) associated with each such group. The methods in (13),(20),(21) do not perform any feature selection. If such selection were coupled to these methods, it should be done in such a way as to account for confounding influences *and* for the fact that confounding variables may have much greater influence on some features than others. A given feature Y_i could be conditionally independent of K given C, independent of C given K, independent of both, or dependent on both but to varying degrees. Moreover, as discussed in section 3.2 for the method in (23), there could be various tied parameter structures e.g. with a customized model only given particular values of C or K, i.e. $P[Y_i|C = \widetilde{c}\,]$. One way to account for unequal confounding influences on particular features is via a (soft) feature partitioning (joint with the sample partitioning), with individual features probabilistically associated to (essentially, given probabilistic membership with) both the class variable and to the confounding variable(s). In this case, rather than maximizing $I(\underline{Y};C)$, one would try to maximize the sum $\sum_{i=1}^{n}\left(P_i I\left(C;Y_i\right)+\left(1-P_i\right)I\left(K;Y_i\right)\right)$, with respect to both the soft data partition and the probabilistic memberships $\{P_i \in [0,1]\}$, which amount to soft feature partitions. A feature j much more strongly influenced by the confounding variable should have $P_j \simeq 0$. This approach would be quite analogous to (23) which involves joint unsupervised clustering and feature selection. However, in this case, the feature selection is guided by "supervision" from the known confounding value. While these ideas are conjectural, they may give a way to extend (13) and (20) so as to embed feature selection.

5.3. Supervised feature selection

While feature selection is an extremely challenging problem in the unsupervised case (where the classes are *a priori* unknown and need to be estimated jointly with their primary features), it is also well-known to be a difficult combinatorial optimization problem even in the supervised case (17), with a number of proposed methods, of varied computational complexity, ranging from greedy search methods to annealing and genetic algorithms. The selection has been done on the basis of supervised criteria, e.g. class separation measures such as Fisher distance and mutual information. More complex techniques build classifiers and then evaluate classification accuracy (e.g. error rates) for numerous candidate subsets of the full feature space, *e.g.* (32),(42). To our knowledge, though, there is little prior work which accounts for confounding variables (with known values on the training set examples) in choosing the features. This problem is discussed in (65). One possibility, borrowing from (20) is to apply standard supervised criteria, but modified to condition on the known confounding values. For example, mutual information is often used, selecting the features Y_i with maximum information about the class variable C, *i.e.*, $I(C;Y_i)$. This can be altered, to select the features with maximum $I(C;Y_i|K)$. However, as we noted in the last section, conditioning entails estimating third order probabilities, whose accuracy may be poor for small microarray training sets. Other criteria such as Fisher distance can likewise be modified by conditioning on K but may suffer from similar problems. The approach proposed in (65), considering the case of multiple patient sites as the confounding influence, is to perform feature selection separately on the data from each site, compare the chosen features across sites, and then *definitively* select the features that are chosen at more than one site (e.g. pick the features deemed informative at least at two out of three sites). This approach should be reliable if there is a sufficient number of samples at each site. However, statistical power is lost by dividing the data into site-specific groups. An alternative scheme proposed here, which does not divide the pooled data, is based on treating the confounding variable (e.g. the patient site) as an *additional* variable to be predicted. In other words, we suggest to form an objective function $P_{ce}+\lambda P_{ke}$, with P_{ce} the empirical count of errors on the training set in predicting the class variable (alternative measures of predictive performance could also be used) and with P_{ke} the count of errors in predicting the *confounding* value. We impose the constraint that a feature can only be used in *one* of the prediction tasks (this can potentially be relaxed to allow *soft* memberships in both prediction tasks). We then learn classifiers (e.g. SVMs or multilayer perceptrons) for both prediction tasks and jointly optimize the partitioning of features between the two tasks, with these optimizations performed to minimize $P_{ce}+\lambda P_{ke}$,. The feature partitioning may be performed e.g. via a locally switching optimization. Here, one first chooses an initial feature partition. One then considers switching a single feature from one task to the other, with the switch retained if it reduces the "sum of task errors" objective function (the predictors will need to be trial-retuned for each trial-switch, to gauge the effect of switching a feature from one prediction task to the other.). One can trial-switch features, cycling through all the features, until there are no further changes. This thus yields a locally optimal feature partition. Features that are more predictive of K than C should be removed by this process. The advantage of this proposed scheme over previously mentioned methods is that it does not compromise statistical power by conditioning on K or by dividing the data into groups for different $K= k$ – *i.e.*, all the data will be used. A potential disadvantage, at least as this procedure is defined above, is that it requires hard-partitioning features to the two prediction tasks – in doing so, some residual predictive power associated with rejected features may be lost.

6. STABILITY OF CLUSTERING SOLUTIONS

6.1. Introduction

Clustering techniques have a long history in the life sciences, with early work including e.g. hierarchical clustering methods applied to numerical taxonomy of species/cladistics (52) and recent work on clustering gene expression data e.g. to identify underlying disease groups (28), to hierarchically organize disease groups (47),(19), to find groups of co-expressed genes (indicative of co-regulation), and to identify patient subgroups with different response to drug treatment. Since clustering results help drive formation of scientific hypotheses, it is extremely important that they be reproducible/robust. In particular, hypotheses should not strongly depend on the particular

realization of measurement noise, on a reasonable level of sample population variability, nor on the particular parameter initialization for the clustering algorithm. Solutions that are robust in this sense are often referred to as "stable" solutions. Unfortunately, many clustering algorithms *are* sensitive to these factors. Partitional clustering methods such as K-means and Expectation-Maximization (EM) for learning Gaussian mixture models converge only to locally optimal solutions, and ones "nearest" to the initial clustering solutions. A poor initialization may yield a poor final clustering result. Moreover, there are in general many local optima and thus, potentially, high solution variance, depending on the scheme used for clustering initialization. Moreover, the degrees of freedom in the solution (and the difficulty in finding good solutions) may increase with increasing data dimensionality and the number of sought clusters. Thus, finding robust, accurate solutions for high-dimensional microarray data is a challenging prospect.

While partitional clustering methods are often used, the most popular approach in life science applications is *hierarchical clustering* (26). This is the approach of choice for inferring hierarchical relationships between classes/groups in forming scientific hypotheses about species similarity, species evolution, and related applications. Yet, ironically, hierarchical clustering methods are well-known to yield results that are highly unstable in the presence of sample variability. The wide use of this inherently unstable approach underscores the need (and the recent impetus in the research literature) for evaluating clustering algorithms and solutions with respect to criteria that capture some notion of stability. There is also clear motivation for incorporating design objectives and algorithmic steps which encourage the formation of stable solutions. In the sequel, we will review stability analysis methods for clustering evalution and design and also discuss some other promising approaches.

6.2. Review of recent methods

A stability criterion gives a measure of solution reproducibility that can be used to compare (and hence to favor) one clustering algorithm/solution with respect to another. One application for such a criterion is in addressing the vexing, longstanding problem of estimating the *number* of clusters present in a given data set (the model order) (27). Here, the stability measure is used to evaluate solutions with varying order. Several such approaches have been proposed in the literature (55),(6),(33). (55) considers a 2-fold cross validation setting, with one fold as "training" and the other as "test". Solutions at the same order are generated for both the training and the test sets. One then evaluates the "prediction strength" of the training solution – for each test cluster, one measures the proportion of data pairs that are also assigned to the same cluster when the training set cluster centers are used to form a solution on the test set. The "prediction strength" at this model order is the *minimum* such proportion, over all test set clusters. The chosen cluster number is the highest model order with prediction strength above a specified threshold. In (55), it was recognized that there is an analogy to the bias-variance dilemma in the prediction strength measure as the model order is increased. (6) developed a similar method, the primary differences lying in the following aspects: 1) in (6), the authors proposed to measure partition similarity based on the sum of i) the number of data pairs in the same cluster in the two partitions *and* ii) the number of data pairs in different clusters in the two partitions; 2) (6) generated numerous pairs of data sub-samplings, with cluster stability evaluated (at each order) based on the cumulative distribution function (cdf) of the paired similarities. The largest model order below which there is a significant transition in the cdf is selected as the chosen order. (6) noted that, unlike penalty function methods, e.g. Bayesian Information Criterion (45), their approach does not require an underlying statistical model for the data. Both (55) and (6) recognized the significance of their approaches for and considered application to hierarchical clustering. The method in (33) is related to (55) and differs from (6) in that the pair of data subsamples is *non-overlapping*. Overlapping subsamples are required in (6) since the similarity measure is evaluated over the overlap subset. However, (33) argues that this may introduce bias in the stability measure since overlapping samples will inherently lead to similar data partitions, at every model order. (33) achieves similarity evaluation with nonoverlapping subsets by effectively building a classifier on one data subsample, to predict the clustering on the second subsample. At each model order, the average prediction accuracy, over numerous data splits, defines the stability measure.

In addition to estimating the number of clusters, other solution parameters can also be optimized via stability criteria. In (7), it was proposed to estimate the feature dimensionality, via the number of principal components, by optimizing a stability criterion. For a small number of clusters, the most stable partitions were achieved by retaining only a few principal components, whereas a larger number of clusters required more features to achieve best stability. In general, some dimension reduction (relative to the full feature dimensionality) always gave the most stable clustering results. Although it has not been experimentally validated, this is suggestive that methods such as (23), which embeds feature (and cluster number) selection within clustering in high dimensions, should yield stable clustering solutions, relative to alternative schemes.

While feature selection can help to "stabilize" clustering in high dimensions, this may be insufficient for algorithms that are inherently unstable such as agglomerative (bottom-up) methods, which start by merging individual data samples. Intuitively, this type of method should be highly sensitive to variations in the data set. Top-down (splitting) algorithms (12), (26),(59) for growing the hierarchy are expected to be more stable since the initial splits involve large subsets of the data and thus should be less dependent on the particular data realization. However, top-down methods are typically greedy algorithms, with no mechanism for "undoing" poor splits at the top of the tree, which may be caused by (poor) cluster initialization. This is a source of instability for top-down hierarchical clustering. There are, however, methods which are top-down and, at the same time, *non-greedy* – in fact,

these methods are essentially insensitive to parameter initialization and seek to find the globally optimal solution. Deterministic annealing for clustering (44) grows a partitional clustering solution, with the number of clusters increasing in a nongreedy fashion via phase transitions in an annealing process, which directly occur so as to minimize a free energy objective function. The cluster bifurcations specify a natural hierarchy of clustering solutions. Deterministic annealing has been demonstrated to give some ability to avoid local minima of the clustering distortion (44). A related approach, but one which additionally enforces a tree-structured partition on the set of learned clusters, was developed in (39). It is expected that these "top-down" annealing methods should be inherently more stable than traditional (both bottom-up and top-down) hierarchical schemes.

Rather than attempting to choose a hierarchical learning algorithm that is inherently stable, an alternative for forming reliable clusterings is to generate a population of unstable solutions and choose, as a stable one, the most "representative" one from this population, such as the population mode. For example, in (19), one of the objectives, in addition to building a classifier for fourteen distinct cancer diseases, is to learn a taxonomic hierarchy of these disease classes. Toward this end, the authors applied their (top-down) tree learning algorithm to numerous data subsamples and then generated a histogram of tree structures. The mode of the histogram is a quite reasonable choice as the most stable tree structure. (19) addressed the (supervised) case where the class labels are known and where the hierarchy consists of a hierarchy of *classes*. In this context, the number of distinct tree structures learned from the different subsamples is in practice fairly limited and a distinct mode of the histogram is expected to be found. An interesting question is how to appropriately extend this approach to the case of purely unsupervised hierarchical learning. In this context, clustering solutions obtained for two different data subsamples will not be identical – at best, the solutions may be fairly similar. Thus, rather than finding the mode of the solution population, the most representative solution should amount to something like a solution "centroid". Finding a stable representative from a family of hierarchical solutions may be investigated in future work.

7. CONCLUSIONS

In this paper, we have identified several emergent, fundamental problems in unsupervised learning/clustering that are highly relevant to applications in bioinformatics and to life sciences in general. We have reviewed recent machine learning approaches for addressing these non-standard problems. To date, there has been only limited investigation of some of these approaches on biosciences data and experimental investigation has not been the focus here. This paper is primarily a review and a "position paper", arguing for increased investigation of these topics. We have also proposed several new ideas, in particular for addressing the confounding variables problem, which we will pursue in future work.

8. REFERENCES

1. O. Alter, P.O. Brown, and D. Botstein: Singular value decomposition for genome-wide expression data processing and modeling. *Proc. National Academy of Sciences*, vol. 97, 10101-10106 (2000)
2. M. Bakay, Z. Wang, G. Melcon, L. Schilta, J. Xuan, P. Zhao, V. Sartorelli, J. Seo, E. Pegoraro, C. Angelini, B. Shneiderman, D. Escolar, Y-W Chen, S.T. Winokur, L.M. Pachman, C. Fan, R. Mandler, Y. Nevo, E. Gordon, Y. Zhu, Y. Dong, Y. Wang, and E.P. Hoffman: Nuclear envelope dystrophies show a transcriptional fingerprint suggesting disruption of Rb-MyoD pathways in muscle regeneration. *Brain*, vol. 129, 996-1013 (2006)
3. J. D. Banfield and A. E. Raftery: Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 803-821 (1993)
4. S. Basu, A. Banerjee, and R. Mooney: Active semi-supervision for pairwise constrained clustering. In *SIAM Intl. Conf. on Data Mining*, 333-344 (2004)
5. Z. Ghahramani and J. M. Beal: Variational inference for Bayesian mixtures of factor analyzers. In *Neural Info. Proc. Systems*, 449-455 (2000)
6. A. Ben-Hur, A. Elisseeff, and I. Guyon: A stability based method for discovering structure in clustered data. In *Proc. Pacific Symposium on Biocomputing*, 6-17 (2002)
7. A. Ben-Hur and I. Guyon: Detecting stable clusters using principal component analysis. *Methods in molecular biology*, 159-182 (2003)
8. M. Benito, J. Parker, Q. Du, J. Wu, D. Xiang, C.M. Perou, and J.S. Marron: Adjustment of systematic microarray data biases. *Bioinformatics*, vol. 20, no. 1, 105-114 (2004)
9.A. Demiriz and K. P. Bennett: Optimization approaches to semi-supervised learning. In *Applications and Algorithms of Complementarity*, Kluwer Academic Publishers, Boston (2000)
10. R.E. Blahut: *Principles and practice of information theory*, Addison-Wesley, Reading, Mass. (1991)
11. A. Blum and T. Mitchell: Combined labeled and unlabeled data with co-training. In *Proc. of Comp. Learning Theory* (1998)
12. A. Buzo, A. H. Gray, R. M. Gray, and J. D. Markel: Speech coding based on vector quantization. *IEEE Transactions on Acoustics., Speech, and Signal Processing*, 562-574 (1980)
13. G. Chechik and N. Tishby: Extracting relevant structures with side information. In *Neural Info. Proc. Systems* (2002).
14. F. G. Cozman and I. Cohen: Unlabeled data can degrade classification performance of generative classifiers. In *Intl. Florida AI Society Conf.*, (2002)
15. N. Cristianini and J. Shawe-Taylor: *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge (2000)
16. A.P. Dempster, N.M. Laird, and D.B. Rubin: Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Stat. Society*, 39(1):1-38, (1977)
17. R. O. Duda and P. E. Hart: *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York, NY (1974)
18. J. G. Dy and C. E. Brodley: Feature selection for unsupervised learning. *Journal of Machine Learning*

Research (2004)
19. Y. J. Feng, Z. Wang, Y. Zhu, J. Xuan, D. J. Miller, R. Clarke, E. P. Hoffman, and Y. Wang: Learning the tree of phenotype using genomic data and VISDA. In *Proceedings of IEEE Workshop on Bioinformatics and Bioengineering* (2006)
20. D. Gondek and T. Hofmann: Non-redundant data clustering. In *IEEE Intl. Conf. on Data Mining* (2004)
21. D. Gondek and T. Hofmann: Non-redundant clustering with conditional ensembles. In *Proc. Knowledge, Discovery and Data Mining* (2005)
22. M. Graham and D. J. Miller: Unsupervised learning of mixtures on huge spaces with integrated feature and component selection. In *Proc. ANNIE* (2004)
23. M. Graham and D. J. Miller: Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection. *IEEE Trans. on Signal Processing*, 1289-1303 (2006)
24. J. Grim: Multivariate statistical pattern recognition with non-reduced dimensionality. *Kybernetika*, vol. 22, no.2, 142-157 (1986)
25. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik: Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389-422 (2001)
26. J. A. Hartigan: *Clustering algorithms*. John Wiley, New York (1975)
27. A. K. Jain and R. C. Dubes: *Algorithms for clustering data*. Prentice Hall, Englewood Cliffs, NJ (1988)
28. D. Jiang, C. Tang, and A. Zhang: Cluster analysis for gene expression data: a survey. *IEEE Trans. on Knowledge and Data Engineering*, 1370-1386 (2004)
29. W. E. Johnson, C. Li, and A. Rabinovic: Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, vol. 8, no. 1, 118-127 (2007)
30. J. Kittler: Feature set search algorithms. In *Proc. Pattern Recognition and Signal Processing*, 41-60 (1978)
31. D. Klein, S. D. Kamvar, and C. D. Manning: From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering. In *Intl. Conf. on Machine Learning*, 307-314 (2002)
32. R. Kohavi and G. H. John: Wrappers for feature subset selection. *Artificial Intelligence*, 273-324 (1997)
33. T. Lange, V. Roth, and J. M. Buhmann: Stability-based validation of clustering solutions. *Neural Computation*, 1299-1323 (2004)
34. M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain: Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1154-1166 (2004)
35. X. Liu, Y. Gong, W. Xu, and S. Zhu: Document clustering with cluster refinement and model selection capabilities. In *ACM Conf. on Res. and Dev. in Info. Retrieval*, 191-198 (2002)
36. S. C. Madeira and A. L. Oliveira: Biclustering algorithms for biological data analysis: a survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 24-45 (2004)
37. G. McLachlan and D. Peel: *Finite mixture models*. John Wiley and Sons, New York (2000)
38. D. J. Miller and J. Browning: A mixture model and EM-based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 1468-1483 (2003)
39. D. Miller and K. Rose: Hierarchical, unsupervised learning with growing via phase transitions. *Neural Computation*, 8(2):425-450 (1996)
40. D. J. Miller and H. Uyar: A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Neural Information Processing Systems*, vol. 9, pp. 571-577 (1997)
41. W. Pan: Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, 795-801 (2006)
42. P. Pudil, F. J. Ferri, J. Novovicova, and J. Kittler: Floating search methods for feature selection with nonmonotonic criterion functions. *IAPR International Conference on Pattern Recognition*, 279-283 (1994)
43. J. Rissanen: Modelling by shortest data description. *Automatica*, vol. 14, 465-471 (1978)
44. K. Rose, E. Gurewitz, and G. C. Fox: Vector quantization by deterministic annealing. *IEEE Trans. Inform. Theory*, 38:1249-1257 (1992)
45. G. Schwarz: Estimating the dimension of a model: *The Annals of Stats.*, vol. 6, no. 2, pp. 461-464 (1978)
46. B. Shashahani and D. Landgrebe: The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. on Geoscience and Remote Sensing*, vol. 32, pp. 1087-1095 (1994)
47. K. Shedden at al.: Accurate molecular classification of human cancers based on gene expression using a simple classifier with a pathological tree-based framework. *American Journal of Pathology*, 1985-1995 (2003)
48. N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall: Computing Gaussian mixture models with EM using equivalence constraints. In *Neural Information Processing Systems* (2003)
49. S. Yu and J. Si: Segmentation given partial grouping constraints. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 173-183 (2004)
50. A. Sierra and F. Corbacho: Reclassification as supervised clustering. *Neural Computation*, 2537-2546 (2000)
51. N. Slonim and N. Tishby: Document clustering using word clusters via the information bottleneck method. In *ACM Conf. on Res. and Devel. in Info. Retr.*, 208-215 (2000)
52. R. Sokal and P. Sneath: *Principles of numerical taxonomy*. W. H. Freenabm San Francisco (1963)
53. A. Strehl and J. Ghosh: A knowledge reuse framework for combining partitionings. *J. of Machine Learning Research*, 583-617 (2002)
54. C. Tang, L. Zhang, A. Zhang, and M. Ramanathan: Interrelated two-way clustering: an unsupervised appraoch for gene expression data analysis. In *IEEE Intl. Symp. on Bioinformatics and Bioengineering* (2001)
55. R. Tibshirani, G. Walther, and T. Hastie: Cluster validation by prediction strength. Stanford Statistics Dept. Technical Report (2001)
56. N. Tishby, F. C. Pereira, and W. Bilek: The information bottleneck method. In *Allerton Conf. on Comm., Control, and Computing*, 368-377 (1999)
57. A. Topchy, M. Law, and A. K. Jain: Combining

multiple weak clusterings. In *Proc. of IEEE International Conference on Data Mining*, 331-338 (2003)
58. K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl: Constrained K-means clustering with background knowledge. In *Intl. Conf. on Machine Learning*, 577-584 (2001)
59. Z. Wang, Y. Wang, J. Lu, S-Y Kung, J. Zhang, R. Lee, J. Xuan, J. Khan, and R. Clarke: Discriminatory mining of gene expression microarray data. *J. VLSI Signal Processing*, 255-272 (2003)
60. caBIG, http://caBIG.nci.nih.gov
61. Y. Wang, L. Luo, M.T. Freedman, and S.Y. Kung: Probabilistic principal component subspaces: a hierarchical finite mixture model for data visualization. *IEEE Trans. on Neural Networks*, vol. 11, 625-636 (2000)
62. E. P. Xing and R. M. Karp: CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 306-315 (2001)
63. E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell: Distance metric learning, with application to clustering with side-information. In *Neural Information Processing Systems* (2002)
64. R. Xu and D. Wunsch II: Survey of clustering algorithms. *IEEE Trans. on Neural Networks*, 645-677 (2005)
65. Z. Zhang and D. W. Chan: Cancer proteomics: in pursuit of "true" biomarker discovery. *Cancer Epidemiological Biomarkers*, 2283-2286 (2005)
66. P. Zhao, J. Seo, Z. Wang, Y. Wang, B. Shneiderman, and E.P. Hoffman: *In vivo* filtering of *in vitro* expression data reveals MyoD targets. *Comptes Rendus Biologies*, vol. 326, 1049-1065 (2003)
67. Q. Zhao and D. J. Miller: Mixture modeling with pairwise, instance-level class constraints. *Neural Computation*, 2482-2507 (2005)
68. Y. Zhu, Z. Wang, Y. Feng, J. Xuan, D.J. Miller, E.P. Hoffman, and Y. Wang: Phenotypic-specific gene clustering using diagnostic tree and VISDA. In *IEEE Intl. Conf. Eng. Med. Biol.*, New York (2006)

Footnotes: [1] A multinomial component model was used for text, instead of a Gaussian model, in (23). The multinomial has one free parameter for each word, per component. [2] Careful study design can minimize confounding influences and sample biases. However, some confounding influences – from patient age, gender, institution, and other factors – may be unavoidable in the measured data.

Abbreviations: BIC: Bayesian Information Criterion, CL: cannot-link, DNA: deoxyribonucleic acid, DWD: distance-weighted discrimination, EB: empirical Bayes, EM: expectation-maximization, IB: information bottleneck, ML: must-link, PCA: principal components analysis, PMF: probability mass function, SVD: singular value decomposition, SVM: support vector machine, VISDA: visual and statistical data analyzer

Key Words: Clustering, Feature Selection, Model Order Selection, Semisupervised Learning, Confounding Effects, Data Fusion, Information Bottleneck, Stability Criteria, Hierarchical Clustering, Review

Send correspondence to: Professor. D.J. Miller, Elec. Engr. Dept, Penn State, University Park, PA, 16802, USA, Tel: 814-865-6510, Fax: 814-865-7065, E-mail: djmiller@engr.psu.edu

http://www.bioscience.org/current/vol13.htm