Removing confounding factors via constraint-based clustering: An application to finding homogeneous groups of multiple sclerosis patients

Jingjing Liu

Department of Computer Science, Tufts University, 161 College Ave, Medford, MA, USA, 02155. jliu02@tufts.edu.

Corresponding Author: Carla E. Brodley

College of Computer and Information Science, Northeastern University, 440 Huntington Avenue 202 West Village H Boston, Massachusetts 02115. c.brodley@neu.edu. Phone: (617) 373-5204

Brian C. Healy

Biostatistics Center, Massachusetts General Hospital, Boston, MA, USA, 02114. bchealy@partners.org.

Tanuja Chitnis

Partners Multiple Sclerosis Center, Brigham and Women's Hospital, Brookline, MA, USA, 02115. tchitnis@rics.bwh.harvard.edu.

Abstract

Objectives: Confounding factors in unsupervised data can lead to undesirable clustering results. For example in medical datasets, age is often a confounding factor in tests designed to judge the severity of a patient's disease through measures of mobility, eyesight and hearing. In such cases, removing age from each instance will not remove its effect from the data as other features will be correlated with age. Motivated by the need to find homogeneous groups of multiple sclerosis (MS) patients, we apply our approach to remove physician subjectivity from patient data.

Methods: We present a method based on constraint-based clustering to remove the impact of such confounding factors. Given knowledge about which feature (or set of features) is a confounding factor, call it F. Our method

first partitions the data into b bins: if F is categorical, instances from the same category construct one bin; if F is numeric, then we split bins such that each bin contains instances of similar F value. Thus each instance is assigned to a single bin for factor F. We then remove feature F from each instance for the remaining steps. Next, we cluster the data separately in each bin. Using these clustering results, we generate pair-wise constraints and then run a constraint-based clustering algorithm to produce a final grouping.
Results: In a series of experiments with synthetic datasets, we compare our proposed methods to detrending when one has numeric confounding factors. We apply our method to the Comprehensive Longitudinal Investigation of Multiple Sclerosis at Brigham and Womens Hospital dataset, and find a novel grouping of patients that can help uncover the factors that impact disease progression in MS.
Conclusions: Our method groups data removing the effect of counfounding factors without making any assumptions about the form of the influence of these factors on the other features. We identified clusters of MS patients that have clinically recognizable differences. Because patients more likely to progress are found using this approach, our results have the potential to aid physicians in tailoring treatment decisions for MS patients.

Keywords:
Constraint-based Clustering, Confounding Factor, Mining Medical Data, Physician Subjectivity, Multiple Sclerosis

---

1. Introduction

Clustering of unsupervised data can help find homogeneous groups in the data. In medical domains, clustering methods have been used to both subtype and identify important features of disease models [1, 2]. For example, we are interested in identifying groups of patients with multiple sclerosis (MS) based on clinical data to understand which patients are most likely to have a more debilitating disease course in order to tailor treatment decisions. However, clustering clinical data is often not straightforward due to two reasons. First, if the patient database is compiled from data from multiple physicians, and some of the features are based on a physician's interpretation of symptoms or tests then we must correct for physician subjectivity. Second, there can be confounding factors in the data that lead to uninformative clustering results and thus need to be removed. For example, age often has

an effect on measures used to judge the severity of a patient's disease (e.g., mobility, eyesight, hearing all are impacted by age). In this paper we present a method for addressing both of these issues via constraint-based clustering.

Clustering algorithms use the feature values of a given data set to identify major patterns or trends without supervisory information, i.e., labels. The goal of clustering is to output the partition of the data that best separates instances according to a similarity metric derived from feature space. For some domains, we have access to additional information or domain knowledge about the types of clusters that are sought in the data. The field of constrained clustering [3] developed out of the need to find ways to incorporate this information when it is available. Indeed, constrained clustering is particularly useful when domain knowledge suggests that the default separation presented in feature space is not the one that is sought. Our research addresses the situation in which we know that we have a confounding factor F in our dataset that if left in the data will lead to undesirable clustering results. F can be a single feature, or a small set of features, but simply removing F is not sufficient because other features are correlated with F (linearly or non-linearly). Before describing our approach, we first provide two grounding domains in which these issues arise.

MS is the most common neurological disease among young adults in the United States, with an overall prevalence of 400,000 [4]. One of the hallmarks of MS is the vast heterogeneity of patients, which ranges from patients who have limited or no disability many years after disease onset (benign MS, [5]) to patients who experience severe disability soon after disease onset (malignant MS, [6]). Given the heterogeneity in disease course as well as the number of treatment options, researchers are interested in finding homogeneous groups of patients in order to determine what causes some patients to progress more rapidly than others. As part of the Comprehensive Longitudinal Investigation of MS at the Brigham and Women's Hospital and Partner's MS Center (CLIMB)[7] we are collecting a longitudinal data set of close to 2000 patients from 15 different doctors. The dataset contains three types of features: demographic, measures extracted from MRI scans using a semi-automated pipeline, and the results from neurological exams. For the latter, the physician performs a standardized neurological examination and assigns scores based on a pre-determined disability scale entitled the Expanded Disability Status Scale (EDSS), which combines eight functional status sub-scores to provide an overal measure of disability [8]. This score is given based on the published algorithm, but the provider's experience and

clinical judgement can impact scoring. Consequently, all clinical exam ratings have a degree of subjectivity. Thus for this dataset, we must remove physician subjectivity from these features to accurately cluster the data.

As a second example, consider the domain of chronic obstructive pulmonary disease (COPD), for which researchers are interested in finding the (sub)-categories of COPD [1]. COPD is a broad disease category defined as an irreversible air flow obstruction [9]. In particular, it is not well understood why two patients of the same age and smoking history differ in their degree of COPD. Clustering has been applied to this domain with the goal of finding homogeneous groups in order to then determine if there are differences in genetic factors among the groups [1, 2]. However, Cho, et al [1] found that when clustering 308 subjects from the NETT Study [10] that cluster separation was weak, and the most distinct cluster had statistically significantly lower values for "pack years" than the others suggesting that the resulting clusters were primarily defined by smoking history rather than genetic factors. Simply removing features such as age and pack years would likely not significantly impact the results as age and pack years are correlated with other features. Thus we conjecture that removing these confounding factors and their correlation from the rest of the features might result in more informative clusters.

In the remainder of this paper we first review existing approaches to removing confounding factors from unsupervised data. We then present a new approach based on constraint-based clustering, describing how constraints are formed that remove the effect of the confounding factor from the final clustering. Because some datasets require that we remove multiple confounding factors, we describe several proposed extensions to handle multiple sets of constraints. We present the results of experiments on synthetic data and on our motivating domain of finding homogeneous subgroups of MS patients. We discuss the medical significance of our preliminary results and conclude with future work.

2. Literature survey

Removal of confounding factors has been studied in both machine learning and statistics. In this section we briefly describe stratification, detrending using mixed effect models and non-redundant clustering and discuss why none is the appropriate approach for our task. In addition, in Section 3.1.3 we discuss related work in constraint-based clustering and in Section 4.3, we

discuss research that addresses physician subjectivity in MS datasets as that is the confounding factor we wish to eliminate in our motivating MS dataset.

2.1. Stratification

If one knows that a dataset has distinct subpopulations, then stratification can be applied to analyze each subpopulation (called *strata*) separately. Stratification is appropriate if the either the strata form distinct "domains" of study or when different models may be suitable for the different strata. Thus in the case of the MS data, we could stratify by physician and analyze the results from each separately. However, for the MS data the number of patients per physician ranges from 74 to 2. For some physicians it is unclear how meaningful such a clustering would be. In our experiments we include only physicians with 20 more patients, but even for 20 patients it may be hard to conclude anything from the clustering results. Thus for our dataset, we require a method for removing the confounding factor that does not require examining the subpopulations individually.

2.2. Detrending

We observe that methods from regression for which we can assume a mixed effect model [11] can be applied to preprocess the data before clustering to remove the "fixed effect" from the data. In this framework we are assuming that the value of each feature in the data f_i is a linear combination of the part we are interested in and some function of the confounding factor (which we call F).[1] This process, also known as detrending, works as follows. Assume that we have a data set of N instances each described by d features, one of which is the variable we want to remove: F. We want not only to remove F before we cluster but also F's contribution to each f_i. A well-known approach is to apply regression as follows: for each feature f_i we construct a model m_i to predict f_i using F: $\hat{f}_i = aF + b$ Once we learn the model m_i we subtract $\hat{f}_i$ (the estimated effect of F) from the observed value of f_i for each instance. At this point, we can now cluster the data in the "F-free" feature space. In essence we are clustering on the residuals. Thus if you know the form of the relationship (e.g., linear, quadratic, exponential) between F and each feature (in order to accurately fit m_i), and that relationship is the same

[1]Note that in a linear mixed effect model, the effect is additive but can be a non-linear function of the confounding factor.

across the entire input space, then this method can remove F's influence on the clustering results. We compare this method as a preprocessing step before clustering to the proposed constraint-based clustering approach. Note that this method is limited to numeric data.

2.3. Non-redundant clustering

Non-redundant clustering methods [12–14] are designed for domains in which one knows that a particular clustering is undesirable, but one does not have explicit knowledge as to the confounding factor(s). Non-redundant clustering was first proposed by Gondek and Hofmann [12]. Given a dataset X in which each data point is described by a vector of discrete features and a corresponding dataset Y of relevance variables, the information bottleneck method (IB) [15] finds a clustering Z of X while preserving information about Y by minimizing the objective function $L = I(X;Z) - \beta I(Z;Y)$, where β is a positive parameter. Conditional information bottleneck clustering (CIB) is an extension of information bottleneck clustering (IB) that allows for *negative* relevance information. Specifically, consider a domain with data X, Y and Z, then the goal of CIB is to find a clustering of X, call this C, that encodes properties of Y in C that cannot yet be reliably inferred based on Z (the undesirable clustering). CIB achieves its goal by minimizing the objective function $F = I(X,C) - \beta I(Y,C|Z)$. However, this method is restricted to discrete data only.

Conditional ensemble clustering (CondEns) [14] makes use of cluster ensemble methods to find a novel, "orthogonal" clustering in the data given an undesirable clustering C of k clusters. The method operates in three stages. First it applies clustering to each of the k clusters in C. Next, it extends each of the these "local" clusterings to the remainder of the data, by assigning each remaining data point (those not in the particular local clustering) to its closest sub-cluster. This produces k clusterings which are then used to form a consensus using a cluster ensemble combination method [16]. The rationale behind CondEns is that by clustering within each group of the undesireable clustering, C, one avoids reproducing the same clustering C. CondEns is similar to our method in that it clusters within partitions of the data. It is different in that we produce pairwise constraints using the local clusterings while CondEns produces global clusterings by building an ensemble from the clusters defined over the dataset by the local clusterings. One disadvantage of CondEns, particularly for small datasets, is the difficulty in producing

high-quality global clusterings by extending the local clusterings defined by small amounts of the data.

Constrained orthogonal average link algorithm (COALA) [13] utilizes the undesired clustering C by adding a cannot-link constraint between every pair of instances that belongs to the same cluster in C. Then COALA effectively increase the distance between two instances $1/\omega$ times if there is a cannot-link constraint between them, where $\omega \leq 1$ is a parameter. It then uses an agglomerative hierarchical clustering algorithm to produce the final clustering while preserving the soft cannot-link constraints.

We could apply non-redundant clustering to our problem as follows. We would leave in our confounding factor, cluster the data, declare that the clustering was undesirable, and then apply non-redundant clustering to find a new "orthogonal" clustering. However this approach is not optimal because we are not seeking to find the clustering most different from some "undesirable" clustering. Consider the case of physician subjectivity in our MS data – even if we cluster the original data this is not necessarily a bad clustering from which we wish to move far away from – aspects of the clustering will be meaningful. Indeed, we will observe that unconstrained clustering is able to distinguish the sick from the not-so-sick MS patients, a distinction that we wish to retain. Finally, it is not clear how to apply non-redundant clustering for multiple undesirable factors.

3. Methods

In this section we first describe how we can remove the effect of a confounding factor using constraint-based clustering. We then extend our method for situations in which there are a small set of possible confounding factors.

3.1. Removing confounding factors via constraint-based clustering

Before describing the details of our approach we provide a brief overview. Recall that we are given knowledge about which feature (or set of features) is a confounding factor, call it F. Our method first partitions the data into b bins: if F is categorical, instances from the same category construct one bin; if F is numeric, then we split bins such that each bin contains instances of similar F value. Thus each instance is assigned to a single bin for factor F. We then remove feature F from each instance for the remaining steps. Next, we cluster the data separately in each bin. Using these clustering results, we

generate pair-wise constraints and then run our constraint-based clustering algorithm.

3.1.1. Binning

The first step is to partition the data into b bins such that data in a particular bin has comparable F values. For categorical F we can define one bin per distinct value of F. For example for the MS data, we have one bin for each physician in the database (recall that there are multiple patients per physician). For numeric F, we can split the data into bins using domain knowledge, or automatically. For example we might split the data into b equal-sized bins or we can split the data into equal-sized ranges. We could also use unsupervised methods such as as non-parametric density estimation [17] or entropy-based discretization [18] to define the bin ranges.

3.1.2. Generating constraints from the bins

After we partition the data into b bins, we next run clustering on each bin separately. Note that we use all features except for F in the clustering. In our experiments we choose to run expectation maximization (EM) [19] over a mixture of Gaussians, but any clustering algorithm could be applied at this step. EM requires that we specify the number of clusters. We can use a criterion such as the Bayesian Information Criterion (BIC) [20] or domain knowledge to choose k_b, the number of clusters for each bin b. From these clustering results we generate the pair-wise "cannot-link" constraints only for instances that are 1) in the same bin and 2) in separate clusters. All other pair-wise constraints are set to be 0. Specifically, if instances x_i and x_j are in the same bin and are in separate clusters then we assign a constraint of -1, indicating that we would like to keep them apart in the final clustering. Note that one could set the constraint between pairs of instances within the same bin and cluster to be 1, indicating that they should be placed in the same cluster. But this is equivalent to requiring that patients from the same physician who end up in the same cluster in a bin must remain together in the final clustering. We prefer to let feature space similarity define which instances should be placed together in the final clustering rather than enforcing the (perhaps erroneous) contraints that similar patients of the same physician be put together in the final clustering.

Our method is based on the intuition that because the instances in a particular bin have comparable F values, the clustering in each bin is independent of F. Consequently, we can then use these constraints to enforce

this structure in the final clustering of the whole data set. The net result is that we remove the effect of F without making any assumptions about the form of the influence of F on the other features.

3.1.3. Constraint-based clustering via CPPC

At this point we run constraint-based clustering on the data with the generated constraints. We choose to use Class-Level Penalized Probabilistic Clustering (CPPC) [21] algorithm. Before describing why this is an appropriate choice for this task, we first review constraint-based clustering and Penalized Probabilistic Clustering (PPC) [22], the algorithm on which CPPC is based.

Constraint-based clustering was originally introduced by Wagstaff, et al [23] as a modification to the k-means clustering algorithm. The constraints were specified as *pair-wise* must-link and cannot-link constraints to incorporate knowledge as to which instance pairs should or should not be clustered together. These "hard" constraints are strictly enforced, requiring any clustering of the data to satisfy all of the pair-wise constraints. Hard constraints may not always be available or even desirable (they can be impossible to satisfy), and thus several algorithms were developed to incorporate probabilistic constraints [22, 24]. In addition, researchers explored how to add constraints to other clustering algorithms such as EM [25–28], hierarchical clustering [29] and spectral clustering [30–34]. In particular, Penalized Probabilistic Clustering (PPC) [22] alters the Gaussian Mixture Model (GMM) to accommodate soft pair-wise constraints by embedding them into the prior over the assignment of instances to clusters. Because inference in this model is intractable, Lu and Leen [22] developed approximations using variational inference as well as Gibbs sampling. CPPC [21] extended PPC to incorporate domain knowledge in the form of group level constraints.

The input for PPC includes $X = \{x_1, x_2, ..., x_N\}$, a dataset of N instances, a set of pair-wise constraints, and the number of clusters k. The constraint between instance x_i and x_j is noted as ω_{ij}. A positive-valued constraint emphasizes the similarity between x_i and x_j, and a negative-valued constraint emphasizes separation. In the E-step, PPC calculates posterior probabilities, q_{ik}, the probability that instance x_i was generated by the k^{th} component given the constraints (ω_{ij}) and the current parameters of the GMM (Θ). In the M-step, PPC uses these probabilities to update Θ. Thus q_{ik} is computed as

follows for PPC:

$$q_{ik} \sim \Pr(Z_i = k|X, \Theta) \exp\left(\sum_{j \neq i} \lambda \omega_{ij} q_{jk}\right) \quad (1)$$

where $\lambda \geq 0$ indicates how much to focus on the constraints versus the feature space. If $\lambda = 0$, PPC is equivalent to EM; if $\lambda = 1$, PPC puts equal weight on the constraints and the feature space. Thus to apply PPC one must specify λ. The time complexity of PPC is $O(kN^2)$.

We choose to use a new variant of PPC called Class-level Penalized Probabilistic Clustering (CPPC) [21] in order to gain a significant speedup over PPC. We are able to use CPPC because of the following observation about our task: if two instances x_i and x_j are in the same bin and the same cluster, the following rule applies:

$$\omega_{il} = \omega_{jl} \; \forall l \in \{1, ..., N\}, l \neq i, l \neq j \quad (2)$$

This means that whatever constraint x_i has with *any* other instance, x_j has the same constraint. Thus we can define x_i and x_j to be in the same *group*. Because we can define a group for each cluster within a bin, the pair-wise constraints between instance pairs become group constraints. This allows us to use CPPC for our constraint-based clustering algorithm, resulting in a significant speedup over PPC [21], which we review here in preparation for our proposed extension of PPC and CPPC to multiple sets of constraints.

In CPPC, the constraint between two instances is a function of their group indicies. More specifically, instead of directly taking a set of constraints, the input for CPPC is a label from one of the L groups for each instance x_i; we denote the group label for x_i as l_i. In addition to a group label, CPPC requires an $L \times L$ matrix C, in which each element $C(l_a, l_b)$ is the value of the probabilistic constraint between group l_a and l_b. CPPC extends PPC to make use of the C matrix by noting that the constraint between a pair of instances x_i and x_j is the corresponding group-wise constraint value $C(l_i, l_j)$. In our application, we set the "label" of an instance to be its cluster and bin, thus we have $\sum_{i=1}^{b} k_{b_i}$ groups, where k_{b_i} is the number of clusters for bin b_i. Recall that in Section 3.1.2, $C(l_i, l_j) = -1$ if l_i and l_j are from the same bin and separate clusters within that bin. In all other cases, $C(l_i, l_j) = 0$.

The time savings of CPPC over PPC comes from the observation that in Equation 1 the summations over all instances for all q_{ik} have many repeated

values because the label of the current instance l_i remains constant, and only changes for the L possible labels of the other instance. Thus we can perform a preprocessing step that requires only $O(kNL)$ time steps while the time complexity for the E-step in EM algorithm is $O(kN)$, and for PPC it is $O(kN^2)$.

$$S(l,k) = \sum_{i=1}^{N} \lambda C(l, l_i) q_{ik} \tag{3}$$

which we compute for each distinct group l and cluster k. We can thus rewrite Equation 1 to be:

$$\begin{aligned} q_{ik} \quad \sim \quad & \Pr(Z_i = k | X, \Theta) \\ & \times \exp\left(2\left(S(l_i, k) - \lambda C(l_i, l_i) q_{ik}\right)\right) \end{aligned} \tag{4}$$

which is calculated in a single step over $N \times k$ possible values. This modification allows each step of the q_{ik} calculation to take constant time, resulting in a time complexity of $O(kNL)$ for CPPC.

3.1.4. On choosing the weight of the constraints

Our method requires that the user specify the value of λ, the parameter that indicates how much to focus on the constraints versus the feature space. Recall that if $\lambda = 0$, PPC is equivalent to EM; if $\lambda = 1$, PPC puts equal weight on the constraints and the feature space. The question then arises of how to set λ for a given dataset. To answer this question we must consider the purpose of clustering.

Clustering is an inherently exploratory process in which a user is interested in determining how instances group together in order to learn something about their data. Indeed, the evaluation of clustering is ultimately subjective as it is up to the domain specialist to determine if a clustering is of interest.[2] Thus we view λ as a "knob" that the user can turn to examine the impact that paying increasing/decreasing attention to the removal of the confounding factor has on the clustering result.

[2]Papers in KDD, ICML and ICDM compare clustering algorithms using measures such as normalized mutual information [35] with the class labels, but in real clustering domains we do not have class labels – for if we did then we would not need to cluster!

3.2. Dealing with multiple confounding factors

Some datasets require that we remove multiple confounding factors. For example, when searching for genetic associations of distinct COPD progression, both age and smoking history are confounding factors because they both are known to influence progression of COPD. In this section we present three ways to extend our method to accommodate small sets of multiple factors.

Our first approach just expands the binning process to create a bin for each unique combination of factors. Without loss of generality consider the case of two confounding factors: we now need a 2-D grid of bins. We can then assign instances to their respective bins, cluster within each bin separately and then use the method described in Section 3.1.3 directly. For example, for the MS data, we could first separate patients by physician and then further separate patients by age – the result would be that each bin contains patients from a single physician and a single age range.[3] With sufficient data and only two confounding factors, *multidimensional binning* merely requires that one can define meaningful bins (note that for larger sets of confounding factors this is highly unlikely). The issue is that for many datasets we will not have sufficient data and indeed, for the MS domain, some physicians see more older/younger patients which would result in insufficient data per bin in these cases.

Our second and third methods are designed to work even with small datasets and multiple confounding factors. To this end, for both we first bin each of the d factors separately to generate d sets of group constraints (note that all d factors are then removed from the data – they are used only to define the bins). The two methods differ in how they use the d sets of group constraints. The first takes the union of the constraints; we define a constraint for a pair of instances, x_i, x_j if *any* of the group constraints defines a constraint for x_i, x_j. We then apply CPPC. The other method keeps the constraints separate and applies CPPC to cluster the whole dataset with multiple group constraints. In the remainder of this section, we first present our modifications to PPC and CPPC to handle multiple group constraints. We then provide an explanation for the rationale behind the approach by examining the impact of multiple factors when they contradict one another for a distinct pair of instances (i.e., removing one factor results in a must-

[3]Note that currently we do not have sufficient data to explore binning by both age and physician.

not-link constraint between the pair of instances and removing the second factor does not).

3.2.1. Modifying PPC and CPPC for multiple sets of constraints

To apply the constraint-based approach for removing *multiple* confounding factors, we extend both PPC and CPPC for multiple sets of constraints. To extend PPC for multiple sets of constraints, we need only change the update function for q_{ik} in the E-step:

$$\begin{aligned} q_{ik} &\sim \Pr(Z_i = k|X, \Theta) \\ &\quad \times \exp\left(\sum_{j \neq i}\left(\sum_{d=1}^{D} \lambda_d \omega_{ijd})\right) q_{jk}\right) \end{aligned} \tag{5}$$

where D is the number of constraint sets, ω_{ijd} is the constraint between instances x_i and x_j for constraint set d, and λ_d specifies the relative importance of each constraint set d. Similarly for CPPC, we need only change the update function for q_{ik}

$$\begin{aligned} q_{ik} &\sim \Pr(Z_i = k|X, \Theta) \\ &\quad \times \exp\left(2\left(\sum_{d=1}^{D} \left(S_d(l_{di}, k) - \lambda_d C_d(l_{di}, l_{di})\right) q_{ik}\right)\right) \end{aligned} \tag{6}$$

where l_{di} denotes the group label of instance x_i in the group constraint from constraint set d. To apply this formulation in our setting of D confounding factors $\{F_1, F_2, ..., F_D\}$, we first bin on each factor F_d and cluster within those bins to generate D sets of group constraints. Each set of constraints has its corresponding constraint matrix C_d for F_d.

3.2.2. Rationale for multi-constraint clustering

Without loss of generality we will examine what it means to have two confounding factors, F_1 and F_2 with respect to the generated constraints. In particular, we explain why combining the constraints from multiple factors makes sense even "when they indicate different things." The insight comes from that we only generate "must-not-link" constraints. Given a pair of instances, x_i and x_j there are three cases:

Case a: For both factors, x_i and x_j are placed in separate bins. In this case no constraints are generated between x_i and x_j.

Case b: x_i and x_j are placed together in a single bin for F_1, but are placed in different bins for F_2. Thus no constraint will be generated for factor F_2. In this case, it is as if we only have one confounding factor, F_1; whether or not a must-not link constraint is generated for x_i, x_j depends on whether they are clustered together in their F_1 bin.
Case c: This case concerns when x_i and x_j end up in the same bin for factor F_1 and are also put together in a single bin for factor F_2. We consider three subcases:

- c-1 If x_i and x_j are clustered together for both factors, then no constraint is generated (indeed in this case, they are close to one another in feature space and likely will end up in the same cluster when all of the data is clustered after defining all of the constraints from F_1 and F_2).
- c-2 If x_i and x_j end up in separate clusters for both factors, then for each factor we generate a must-not-link constraint. Thus we want to keep them far apart in the final constrained clustering.
- c-3 If x_i and x_j are clustered together in the bin for F_1 but are in different clusters in the bin for F_2, then one must-not-link constraint will be generated for F_2. Thus some evidence exists that they should be kept separated but its influence will depend on λ_2.

Thus there are only two situations in which only one of the factors F_d generates a "must-not-link" constraint (cases b and c-3) for a given pair of instances x_i, x_j. In both cases, partial evidence exists that suggests that they should not be placed in the same final cluster. How much attention this is given depends on the weight given (λ_d) for factor F_d. Note that the combination of multiple factors works because our method generates only "must-not-link" constraints. Consider an approach that did generate "must-link" constraints for each pair of instances that are found in the same cluster in the same bin. Then we could have a situation in which a pair of instances might have both a "must-link" **and** a "must-not-link" constraint. Thus they would cancel out and neither factor would be removed. But as described in Section 3.1.2, "must-link" constraints are unnecessary and thus we never have the case of conflicting constraints for a pair of instances.

4. Results

In this section we present three sets of experiments. The first uses synthetic data to compare our proposed method to detrending using regression

when one has numeric confounding factors. The second compares several methods for removing multiple confounding factors. Finally the third experiment is based on our motivating domain of MS, in which we apply our proposed method to remove physician subjectivity from MS data.[4]

4.1. Experiments on synthetic data

We present experimental results on three synthetic datasets. Each is generated by adding a confounding factor to the image segmentation dataset from the UCI Machine Learning Repository [36] which has $N = 2310$ instances, each described by $m = 19$ features. Note that this data is supervised but we ignore the label during clustering. We denote the features as $\{f_1, ..., f_m\}$. We create three datasets, each with a single confounding factor, by first generating a number F_i between 0 and 1 for each instance $i \in \{1, ..., N\}$ using a uniform distribution. F is our confounding factor, which we use to generate the synthetic datasets as follows:
Linear: F affects all features linearly. The j^{th} feature of the i^{th} instance is generated by: $f'_{i,j} = f_{i,j} + c_j F_i$, where c_j is a feature-specific positive constant. For our experiments $c_j = j$; having a different constant for each feature means that F affects each feature by a different magnitude.
Quadratic: F has quadratic effect on features. The j^{th} feature of the i^{th} instance is generated by: $f'_{i,j} = f_{i,j} + c_j F_i^2$, where $c_j = j$.
Logarithmic: F has logarithmic effect on the features. The j^{th} feature of the i^{th} instance is generated by: $f'_{i,j} = f_{i,j} + c_j \log F_i$, where $c_j = j$.

In our experiments, for each method compared, we run the clustering for ten different randomly chosen starting cluster centers. In all cases we set the number of clusters k to be seven, which is the number of classes in the supervised version of the segmentation dataset. In each case we present the result that maximizes the likelihood over the ten runs (note that for CPPC this is the pseudo-likelihood function). We report performance of each method based on their normalized mutual information (NMI) [35] with the original class labels. Thus we wish to maximize NMI.

To implement the proposed method, we must choose the number of bins and the method of binning. For this experiment we arbitrarily chose three bins. Because we have domain knowledge that F was generated by a uniform

[4]In all our experiments, the CPPC algorithm is implemented in C++ and source code is available upon request.

distribution over $[0, 1]$, we split F to equal sized intervals. To generate the constraints, we chose to also set k_b the number of clusters for each bin b to be seven.[5] CPPC requires that we specify the value of λ, the parameter that indicates how much weight to put on the constraints in Equation 2. For our experiments we chose a value of 1.0 indicating that we consider the constraints equally important to the feature values. As discussed in Section 3.1.4, λ is best set by the domain expert – clustering is an exploratory data analysis technique and the expert may wish to examine the results for several different values of λ to determine which clustering result is of most interest.

We compare our results to EM, and EM after pre-processing the data by detrending using regression (described in Section 2) and to a piecewise regression method that we describe next. Recall that detrending works well if we know the underlying model of how our factor impacts our features and is the same for all parts of the feature space. For tasks for which either or both of these assumptions is not met, we can compute a piece-wise linear approximation. In this case we can bin the data similar to our constraint-based approach and then apply detrending separately to each bin. Thus in the following experiments we compare this method to the proposed constraint-based method using the exact same bins.

In Figure 1 we show the results for the three synthetic datasets. For each dataset, we compare the results for EM, our proposed constraint-based approach and detrending using regression. In addition, we show as a baseline, the NMI of EM applied to original segmentation dataset; i.e., the dataset without any confounding factors. We show three variants of detrending that differ in the underlying model: a linear model, the optimal model for the dataset and a piece-wise linear model. Note that the "optimal" models for the quadratic and logarithmic datasets are a quadratic and a logarithmic model respectively. The y-axis shows the NMI with the original supervised labels in the dataset. All methods perform significantly better than just applying EM. In all cases, the constraint-based approach obtains the best performance with regression using the optimal model coming in second. Regression performs slightly worse because it removes all variance along the direction of the confounding factor F, which may include meaningful variance from the original features.

[5] We experimented with other values of k_b and found that the results were the same for values of $k_b = 5, 6, 7, 8, 9$.

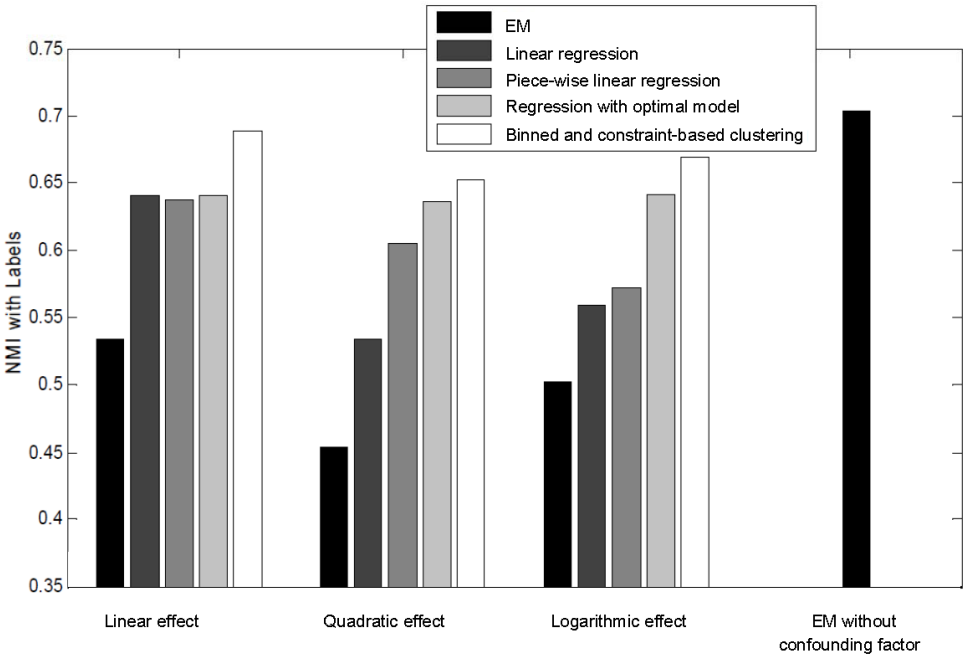


Figure 1: Comparison of EM, detrending with regression and the constraint-based method.

4.2. Multiple confounding factors

To examine how well our method handles multiple confounding factors, we created a fourth synthetic dataset which has two confounding factors. To this end, we first generate two numbers F_{1i} and F_{2i} between 0 and 1 for each instance i using a uniform distribution. The **2-Factor** dataset is created such that the j^{th} feature of the i^{th} instance is generated by: $f'_{i,j} = f_{i,j} + F_{1i} + F_{2i}$. In our experiments we compare the three proposed methods (presented in Section 3.2) of dealing with multiple confounding factors to both EM on the original unmodified data (our gold standard) and on the data after it has been modified for the two confounding factors. In addition we also present results of applying multivariate regression [37] to estimate a single linear model for both F_1 and F_2. For the multiple constraint approach we used three bins for each of F_1 and F_2. Thus to ensure an apples to apples comparison, we used nine uniform bins to form the three by three grid using the same ranges as the bins for the union and multiple constraint approaches. For all methods,

we chose seven clusters for each bin. For multidimensional binning and the union-constraint methods we set $\lambda = 1.0$; for the multiple constraints method we set $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, putting equal weight on both factors. The results are shown in Figure 2.

We see that EM on the original data performs best (rightmost bar). Both linear regression and piece-wise linear regression perform poorly. Indeed even more poorly than EM on the modified dataset. This is because there are two confounding factors present, and linear regression cannot distinguish multiple factors. Multi-dimensional binning (fourth bar from left) performs slightly better than the multiple constraint method (fifth bar from left), which is to be expected since multi-dimensional binning considers both factors together. However, to apply multi-dimensional binning in practice requires that we have sufficient data per bin, which may not always be possible, particularly for larger sets of confounding factors. In such cases, the multiple constraint method can provide a close approximation. The union-based approach performed slightly worse than the multiple constraint approach because it does not distinguish between pairs of instances for which there is some evidence to keep them apart (a constraint generated by one of the two factors) from those where there is stronger evidence (i.e., constraints from both factors).

4.3. Removing physician subjectivity from multiple sclerosis data

As briefly described above, the CLIMB Study is a longitudinal study of MS patients. At entry to the study, patients have a detailed MS history taken. Patients have a complete neurological exam every six months, and a brain MRI every year, which is segmented to measure the brain parenchymal fraction (BPF) and lesion volume. In addition to this information, any clinical relapses or treatment changes that occur during follow-up are recorded by the physician.

One difficulty in MS clinical research is the physician's subjectivity of rating patients in terms of disease severity. The intra/inter-rater variability in the EDSS has received the most attention in the literature [38, 39]. However despite this it continues to be well-accepted by regulatory agencies as the primary or secondary outcome measure in MS. This variability confronts all of the clinical exam measurements. Because MS is a heterogeneous disease, the variability in measuring disease severity added by physician subjectivity is potentially obscuring the impact of treatments or other factors on disease course. Given this issue, many clinical trials require the same physician to examine each patient for the duration of the trial to eliminate the chance of

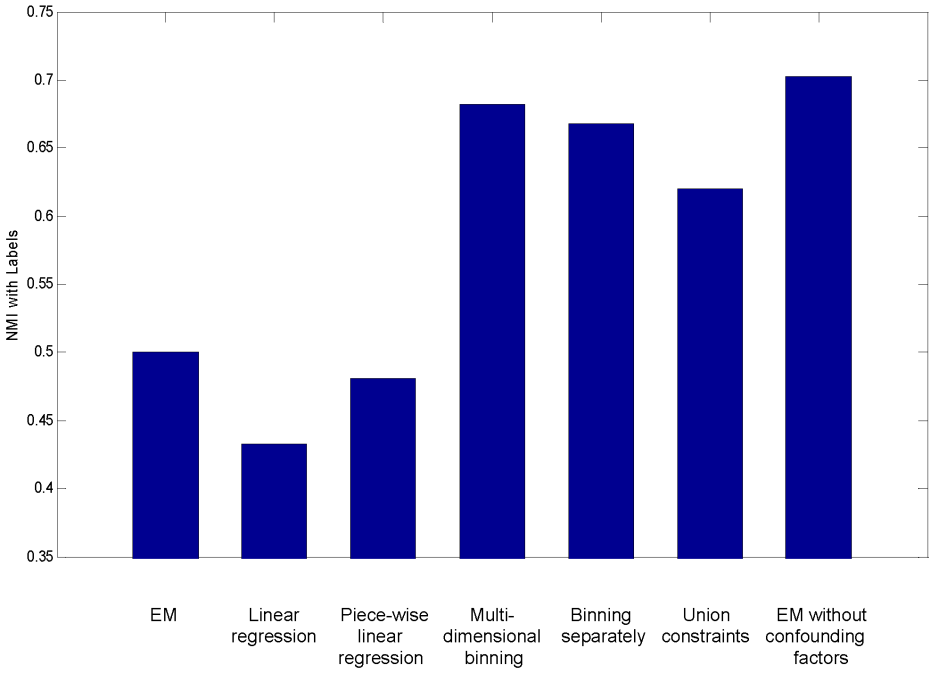


Figure 2: Comparison of EM, detrending with regression and the two methods of dealing with multiple confounding factors.

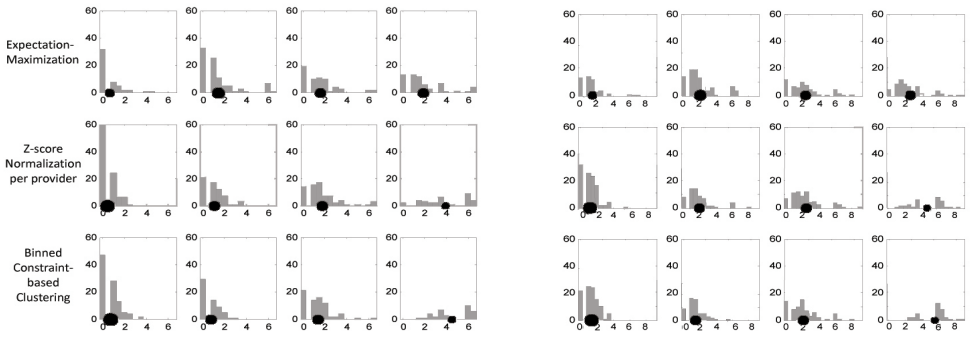
a false positive result on the intervention. However, use of a single physician is often not possible in longitudinal natural history studies which last more than 12-24 months. Longitudinal studies are necessary to provide new insights into the progressive phases of MS for which there are currently no treatments. A new scoring algorithm for the clinical exam (NeruoStatus, http://www.neurostatus.net) has been developed to potentially remove subjectivity and is in use at the Partners MS Center, but this too is subject to physician variability [40]. Therefore, removing variability caused by physician subjectivity could potentially improve clustering and other aspects of clinical research.

We compare the proposed constraint-based method to doing nothing (EM) and to an approach that first preprocesses the data to attempt to remove physician subjectivity through z-score normalization. To this end,

we split the data into p separate datasets, one for each of our p physicians, and normalize within each physician dataset. We then run EM on the entire set of (now normalized) data. Because there are 95 distinct features and only 266 patients clustering in the original space does not lead to meaningful clustering results; clustering with original features suffers from the *curse of dimensionality*: with fixed number of data points, as the number of dimensions in a dataset increases, distance measure becomes meaningless. Techniques for clustering high dimensional data have included both feature transformation and feature selection techniques [41]. In this work, we choose the former method. For all three approaches we perform a principle component analysis (PCA) and retain the first 26 principal components, which explain 90% of the variance in the data. Note that for the normalized data, we perform PCA after normalization. For all three methods, we set the number of clusters k to be four based on our physicians' intuition about how many distinct groups of MS patients there are.

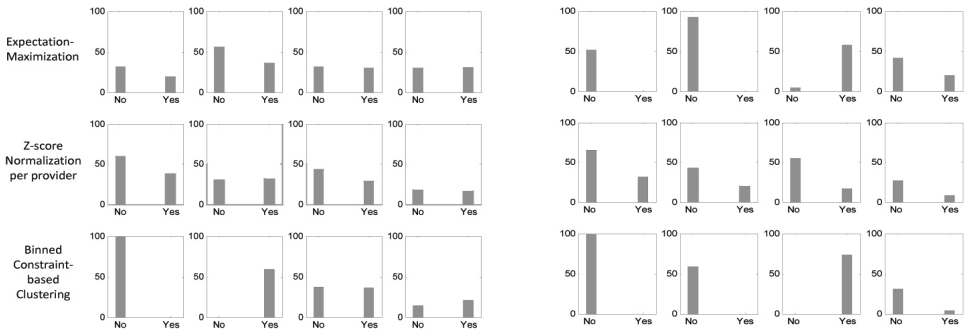
Recall that MS researchers are interested in what are the distinguishing features of groups of MS patients that have different disease course or different response to therapy. This information would aid in choosing optimal treatment strategies and in providing accurate prognosis for individual patients. Thus we want to analyze the clusters from each method to see differences in the distribution of key features of interest. Our study currently contains 530 patients for whom we have three years of data. We omitted those patients whose MS symptoms improved in the three year period as for this project we are interested in those whose disease progresses. In particular we wished to distinguish between patients who progress at slower/faster rates no matter what medication they are given. In addition, we removed patients from doctors with fewer than 20 patients (we conjectured that fewer than 20 patients in a bin does not allow us to cluster the data in that bin in a meaningful way). Thus we are left with 266 patients from seven physicians to cluster. Note that we cluster on only the data at time period 0 when the patient entered the study as we are primarily interested in understanding the reasons for differences in disease progression.

There are too many features to show complete results. Instead we highlight several points of interest that lead our MS physician team members to prefer the constraint-based clustering over the EM or z-score normalization



(a) EDSS month 0

(b) EDSS month 36



(c) Smoking ever

(d) Family history of MS

Figure 3: Multiple Sclerosis results

clusterings.[6] In Figures 3(a) and 3(b) we show the distribution of the EDSS values for time period 0 (when the patient entered the CLIMB study) and time period 36 (three years into the study). Note that neither feature was used to cluster the data. The center position of black ball marks the feature mean value for the cluster, and the size of black ball is proportional to the size of cluster (see Table 1 for the size of each cluster for each of the three methods). In each figure, the clusters are always in the same order within a

[6]Team members were shown the historgrams of of each feature for each cluster to make this analysis. In future workwe will address what is a good interface for showing experts such clustering results.

Method	1	2	3	4
Expectation Maximization	51	93	60	62
Z-Score Normalization per Provider	96	64	58	38
Binned Constraint-Based Clustering	99	57	76	34

Table 1: The size of each cluster for each of the three cluster approaches – shown in the same order as in Figure 3.

clustering method and our initial sorting was based on the cluster mean value of EDSS at time 0. Recall that EDSS is an overall measure of MS patient health. In Figures 3(c) and 3(d), we show the distribution of patients in each cluster that have a history of smoking and family history of MS. Note that both features are included in the 95 features used to compute the principal component representation of the data.

5. Discussion

The results for the synthetic data show a clear win for the proposed constraint-based clustering approach. To understand why it performs better than either EM or EM applied to z-score normalized data on the MS data requires a deeper analysis.

Examining the figures for EDDS 0 and EDDS 36, we see two important results. First, both the new approach and zscore-normalization are much more effective than the EM approach at identifying patients who are the most severe at baseline (cluster 4). For both results, cluster 4 is characterized by having the highest EDSS values, and these patients would be important to investigate more carefully to assess if they have a phenotype similar to malignant patients previously identified by our group. [42].

Second, both the new approach and zscore-normalization were able to identify a subset of patients with mild disease at baseline who were more likely to progress over the three years of follow-up (cluster 3 for both the zscore and the constraint-based approaches).[7] These groups of patients are particularly interesting because they would be candidates for more aggressive

[7]A significant number of patients moved from having an EDSS score below 2 in time period 0 to above 2 in time period 36. An EDSS of 2 is a significant disability threshold in MS.

therapy even though their initial disease course is not dissimilar from the other two mild-disease clusters (clusters 1 and 2).

Why the physicians on our team prefer the new approach over z-score normalization can partially be seen by examining Figures 3(c) and 3(d), which show the distribution of patients in each cluster that have a history of smoking and family history of MS respectively. Note that both features are included in the 95 features used to compute the principal component representation of the data. The new approach provides more separation in these features among the clusters than z-score normalization. These are particularly interesting results for two reasons. First, given the relationship between disease course and smoking [6], these results may point to characteristics of these groups that would not have been identified using traditional approaches. The difference between clusters 1 and 2 for the constraint-based approach is that patients in cluster 2 smoke and thus we conjecture that they progress slightly more at the end of the three year period due to smoking. Second, the difference between clusters based on family history may point to a genetic effect on disease progression which may be independent of smoking history. Although no consistent genetic marker for disease severity in MS has been found, our results demonstrate that patients with a family history are overrepresented in cluster 3 (for the constraint-based approach), which might indicate an underlying genetic trait that is associated with disease progression. This is particularly interesting because it was previously thought [43–46] that a family history of MS had little effect on the disease course of MS. In our results we see that for the constraint-based method, cluster 3 progresses more than cluster 2 (the mean EDSS at time point 36 for cluster 3 has moved above 2.0) and that all of the patients in cluster 3 have a family history of MS whereas all of the patients in cluster 2 do not. What is additionally interesting about this result is that other measures that are thought to have a large impact on progression, such as age of disease onset (i.e., when the patient first showed symptoms of MS), are pretty much indistinguishable between clusters 2 and 3. This separation in a family history of MS is not found by clustering after z-score normalization.

We conjecture that z-score normalization doesn't perform as well as the constraint-based approach because doctors do not see a normal distribution of patients let alone the same distribution. Indeed, older patients tend to see older doctors and some doctors see patients with more/less disease progression. Additionally, physician subjectivity is not necessarily linear, in that they may be less/more subjective for some measures and the direction of

their subjectivity can differ.

Based on these results the next steps are to further evaluate the effect of family history of MS on the disease course. Specifically, we will evaluate the effect of first versus secondary degree relatives, as well as known MS-risk alleles which may modify the disease course. This work may aid in identifying groups of patients at higher risk for disability accrual.

6. Conclusions

In this article we have presented a novel approach to removing confounding factors when clustering data. Our approach first bins the data using the confounding factor F, and then clusters the data within each bin in order to generate constraints that are subsequently used to constrain clustering of the entire dataset. Because instances in a particular bin have comparable F values, the clustering in each bin is independent of F. Consequently, we can then use these constraints to enforce this structure in the final clustering of the whole data set. The net result is that we remove the effect of F without making any assumptions about the form of the influence of F on the other features. In addition to a new approach to removing confounding factors from data, we extended the PPC and CPPC algorithms to handle multiple sets of constraints and we applied our proposed approach to MS data to identify clusters of patients that have clinically recognizable differences. Because patients more likely to progress are found using this approach, our results could aid physicians in tailoring treatment decisions for MS patients. Indeed, the next step is to investigate the genetic and immunological characteristics of these patients to determine the reason behind the difference between clusters in order to understand patient heterogeneity in MS.

7. Acknowledgements

Carla Brodley and Jingjing Liu were supported by support from the NSF (IIS-0803409) and by a pilot research aware from the National Mulitple Sclerosis Society (PP1873). Tanuja Chitnis and Brian C. Healey were supported in part by grants from the National MS Society (RG-4256A4/2).

References

[1] M. Cho et al, G. Washko, T. Hoffmann, G. Criner, E. Hoffman, F. Martinez et al, Cluster analysis in severe emphysema subjects using phenotype and genotype data: An exploratory investigation, Respiratory Research (2010) 30–38.

[2] M. Paoletti, G. Camiciottoli, E. Meoni, F. Bigazzi, L. Cestelli, M. Pistolesi, C. Marchesi, Explorative data analysis techniques and unsupervised clustering methods to support clinical assessment of COPD phenotypes, Journal of Biomedical Informatics (2009) 1013–1021.

[3] S. Basu, I. Davidson, K. Wagstaff (Eds.), Constrained Clustering: Algorithms, Applications and Theory, Chapman & Hall/CRC, 2008.

[4] A. Compston, I. McDonald, J. Noseworthy, H. Lassmann, V. Vienna, D. Miller et al, McAlpine's Multiple Sclerosis. 4th edition, Churchill Livingstone, 2005.

[5] G. S. Ramsaransing, J. D. Keyser, Benign course in multiple sclerosis: A review., Acta Neurologica Scandinavica (2006) 113(6):359–369.

[6] B. Healy, E. Ali, C. Guttmann, T. Chitnis, B. Glanz, G. Buckle et al, Smoking and disease progression in multiple sclerosis, Archives of Neurology (2009) 858–864.

[7] S. Gauthier, B. Glanz, M. Mandel, H. Weiner, A model for the comprehensive investigation of a chronic autoimmune disease: The Multiple Sclerosis CLIMB study, Autoimmunity Reviews (2006) 5(8):532–6.

[8] J. Kurtzke, Rating neurologic impairment in multiple sclerosis: An Expanded Disability Status Scale (EDSS), Neurology (1983) 33(11):1444–52.

[9] J. Vestbo, S. Hurd, A. Agusti, P. Jones, C. Vogelmeier, A. Anzueto, Global strategy for the diagnosis, management, and prevention of COPD: GOLD executive summary., American Journal of Respiratory and Critical Care Medicine (2007) 532–555.

[10] G. Criner, F. Cordova, A. Sternberg, F. Martinez, National emphysema treatment trial, Proceedings of the American Thoracic Society (2008) 393–405.

[11] D. J. Yanosky II, Comparability of covariance structures and accuracy of information criteria in mixed model methods for longitudinal data analysis, Ph.D. thesis, The University of Georgia (2007).

[12] D. Gondek, T. Hofmann, Conditional information bottleneck clustering, in: International Conference on Data Mining, workshop on clustering large data sets, IEEE Computer Society, Los Alamitos, CA, USA, 2003, pp. 36–42.

[13] E. Bae, J. Bailey, COALA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity, in: International Conference on Data Mining, IEEE Computer Society, Los Alamitos, CA, USA, 2006, pp. 53–62.

[14] D. Gondek, T. Hoffmann, Non-redundant clustering with conditional ensembles, in: International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2005, pp. 70–77.

[15] N. Tishby, F. C. Pereira, W. Bialek, The information bottleneck method, Allerton Conference on Communication, Control and Computing (1999) 368–377.

[16] A. Strehl, J. Ghosh, Cluster ensembles: A knowledge reuse framework for combining partitionings, Journal of Machine Learning Research (2003) 583–617.

[17] M. Biba, F. Esposito, S. Ferilli, N. Di Mauro, T. Basile, Unsupervised discretization using kernel density estimation, in: International Joint Conference on Artificial Intelligence, AAAI Press, 2007, pp. 696–701.

[18] D. Chiu, A. Wong, B. Cheung, Information discovery through hierarchical maximum entropy discretization and synthesis, in: International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 1991, pp. 125–140.

[19] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B (1977) 1–38.

[20] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics (1978) 461–464.

[21] D. Preston, C. Brodley, R. Khardon, D. Sulla-Menashe, M. Friedl, Redefining class definitions using constraint-based clustering: An application to remote sensing of the Earth's surface, in: International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2010, pp. 823–832.

[22] Z. Lu, T. K. Leen, Penalized probabilistic clustering, Neural Computation.

[23] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained k-means clustering with background knowledge, in: International Conference of Machine Learning, Morgan Kaufmann Publishers Inc., 2001, pp. 577–584.

[24] S. Basu, A probabilistic framework for semi-supervised clustering, in: International Conference on Knowledge Discovery and Data mining, ACM, New York, NY, USA, 2004, pp. 59–68.

[25] N. Shental, A. Bar-hillel, T. Hertz, D. Weinshall, Computing gaussian mixture models with em using equivalence constraints, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge MA, USA, 2004, pp. 465–472.

[26] E. P. Xing, A. Y. Ng, M. I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge MA, USA, 2002, pp. 505–512.

[27] Z. Lu, M. A. Carreira-Perpinan, Constrained spectral clustering through affinity propagation, in: Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, USA, 2008, pp. 1–8.

[28] B. Kulis, S. Basu, I. Dhillon, R. J. Mooney, Semi-supervised graph clustering: A kernel approach, in: International Conference on Machine Learning, Springer, 2005, pp. 457–464.

[29] I. Davidson, S. S. Ravi, Agglomerative hierarchical clustering with constraints: Theoretical and empirical results, in: European Conference on Principles and Practice of Knowledge Discovery in Databases, Springer, Berlin Heidelberg, 2005, pp. 59–70.

[30] A. Li, J. Walling, S. Ahn, Y. Kotliarov, Q. Su, M. Q. et al, Unsupervised analysis of transcriptomic profiles reveals six glioma subtypes, Cancer Research (2009) 69(5), 2091–2099.

[31] L. C. Jiao, F. Shang, F. Wang, Y. Liu, Fast semi-supervised clustering with enhanced spectral embedding, Pattern Recognition 45 (12) (2012) 4358–4369.

[32] X. Ji, W. Xu, Document clustering with prior knowledge, in: International Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2006, pp. 405–412.

[33] Z. Lu, M. A. Carreira-Perpinán, Constrained spectral clustering through affinity propagation, in: Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, USA, 2008, pp. 1–8.

[34] Q. Xu, M. Desjardins, Constrained spectral clustering under a local proximity structure assumption, in: International Conference of the Florida Artificial Intelligence Research Society, IAAA Press, Menlo Park, CA, 2005, pp. 866–867.

[35] S. Kullback, Information Theory and Statistics., Dover, 1997.

[36] M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

[37] S. Chatterjee, A. Hadi, Influential observations, high leverage points, and outliers in linear regression, in: Statistical Science, Institute of Mathematical Statistics, 1986, pp. 379–436.

[38] M. Hohol, E. Orav, H. Weiner, Disease steps in multiple sclerosis: A simple approach to evaluate disease progression., Neurology (1995) 45(2):251–255.

[39] J. Hobart, J. Freeman, A. Thompson, Kurtzke scales revisited: The application of psycometric methods to clinical intuition., Brain (2000) 123(5):1027–1040.

[40] M. D'Souza, O. Yaldizli, E. Lucassen, E. Kornyeyeva, L. Kappos, The effect of standardised EDSS re-training on the performance of EDSS

raters, in: Multiple Sclerosis Journal, Sage Publications, 2012, pp. 18:339–339.

[41] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: A review, SIGKDD Explorations (2004) 73:90–105.

[42] T. Gholipour, B. Healy, N. Baruch, H. Weiner, T. Chitnis, Demographic and clinical characteristics of malignant multiple sclerosis., Neurology. (2011) 76(23):1996–2001.

[43] E. Celius, H. Harbo, T. Egeland, F. Vartdal, B. Vandvik, A. Spurkiand, Sex and age at diagnosis are correlated with the hla-dr2, dq6 haplotype in multiple sclerosis., Journal of the Neurological Sciences (2000) 178(2):132–135.

[44] A. Hensiek, S. Sawcer, R. Feakes, J. Deans, A. Mander, E. Akesson et al, HLA-DR 15 is associated with female sex and younger age at diagnosis in multiple sclerosis., Journal of Neurology, Neurosurgery and Psychiatry (2002) 72(2):184–187.

[45] T. Masterman, A. Ligers, T. Olsson, M. Andersson, O. Olerup, J. Hillert, HLA-DR15 is associated with lower age at onset in multiple sclerosis., Annalysis of Neurology (2000) 48(2):211–219.

[46] S. Al-Shammari et al, HHV-6 DNAaemia in patients with multiple sclerosis in kuwait, Acta Neurologica Scandinavica (2003) 107:122–124.