Unifying Dependent Clustering and Disparate Clustering for Non-homogeneous Data

M. Shahriar Hossain[1], Satish Tadepalli[1], Layne T. Watson[1],
Ian Davidson[3], Richard F. Helm[2], Naren Ramakrishnan[1]
[1]Dept. of Computer Science, [2]Dept. of Biochemistry, Virginia Tech, VA 24061, USA
[3]Dept. of Computer Science, UC Davis, CA 95616, USA
Email: {msh, stadepal, ltw}@cs.vt.edu, davidson@cs.ucdavis.edu, helmrf@vt.edu, naren@cs.vt.edu

ABSTRACT

Modern data mining settings involve a combination of attribute-valued descriptors over entities as well as specified relationships between these entities. We present an approach to cluster such non-homogeneous datasets by using the relationships to impose either dependent clustering or disparate clustering constraints. Unlike prior work that views constraints as boolean criteria, we present a formulation that allows constraints to be satisfied or violated in a smooth manner. This enables us to achieve dependent clustering and disparate clustering using the same optimization framework by merely maximizing versus minimizing the objective function. We present results on both synthetic data as well as several real-world datasets.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications - Data Mining; I.2.6 [Artificial Intelligence]: Learning

General Terms: Algorithms, Measurement, Experimentation.

Keywords: Clustering, relational clustering, contingency tables, multi-criteria optimization.

1. INTRODUCTION

This paper focuses on algorithms for mining non-homogeneous data involving attribute-valued descriptors over objects from different domains and connected through a relationship. Consider, for instance, the schematic in Fig. 1 (top) which reveals many-many relationships between companies and countries. Each company is characterized by a vector indicating stock values, profit margins, earnings ratios, and other financial indicators. Similarly, countries are characterized by vectors in a different space, denoting budget deficits, inflation ratio, unemployment rate, etc. Each company is also related to the countries that it conducts business in.

Since Fig. 1 (top) has two different vector spaces and one relation, there can be diverse objectives for clustering such a non-homogeneous dataset. We study two broad objectives here. In Fig. 1 (bottom left), we seek to cluster companies (by their financial indicators) and cluster countries (by their economic indicators) such that the relationships between individual entities are **preserved** at the cluster level. In other words, companies within a cluster tend to do business exclusively with countries in a corresponding cluster. In Fig. 1 (bottom right), we identify clusters of companies and clusters of countries where the original relationships between companies and countries are actually **violated** at the cluster level. In other words, clusters in the company space tend to do business with (almost) all clusters in the country space. These two conflicting goals of clustering are meant to reflect two competing hypotheses about companies and their economic performances:

1. **Dependent clustering:** Fortunes/troubles of individual companies are inter-twined with the fortunes and woes of the countries they do business in. This school of thought would support the contention that General Motors's (GM) financial troubles began with the collapse of the mortgage industry in the United States.

2. **Disparate clustering:** Diversification helps prepare companies for bad economic times, and hence performance of companies may not necesarily be tied to (and is, hence, independent of) country indicators. An oft cited example here is that Google is well positioned to weather economic storms because its advertisers were broad based.

Observe that in either case, the clusters are still local in their respective attribute spaces, i.e., points within a cluster are similar whereas points across clusters are dissimilar.

Without advocating any point of view, we posit that it is important to design clustering algorithms that can support both the above analysis objectives. The need for clustering non-homogeneous data with such conflicting criteria arises in many more contexts, including bioinformatics (studied here), social networks [15], hypertext modeling, recommender systems, paleontology, and epidemiology.

The chief contributions of this paper are:

- An integrated framework for clustering that unifies dependent clustering and disparate clustering for non-homogeneous datasets. Unlike prior work that views constraints as boolean criteria, we present a formulation that allows constraints to be satisfied or violated in a smooth manner.

KDD'10, July 25–28, 2010, Washington, DC, USA.

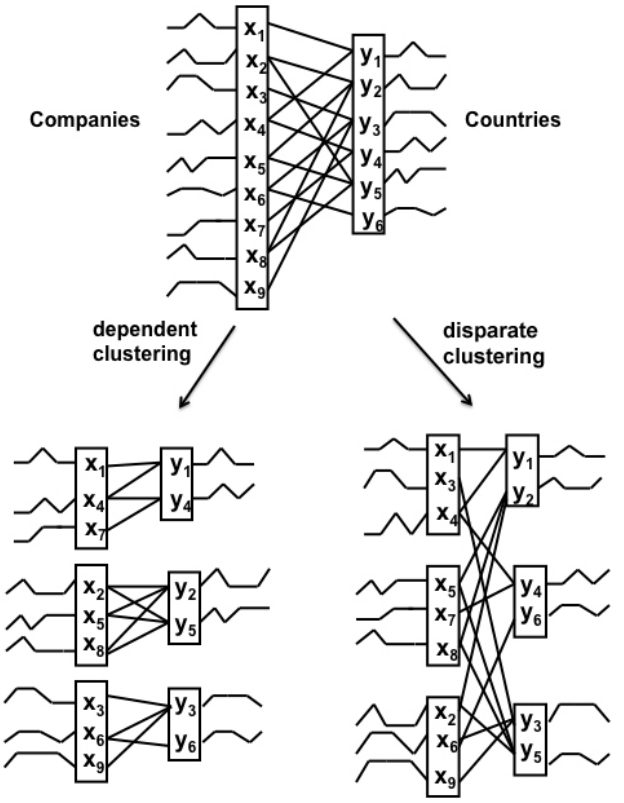

Figure 1: Clustering non-homogeneous data with two different criteria. Here both domains are clustered into three clusters each based on their attribute vectors. (left) Dependent clustering. (right) Disparate clustering.

- While the idea of dependent clustering through a relation has been studied previously [2], the idea of disparate clustering where the objects are of different spaces has not been studied before. We propose this problem here and, moreover, show how we can view dependent clustering and disparate clustering as two sides of the same coin. We propose an integrated objective function whose minimization or maximization leads us to disparate or dependent clustering (resp.)

- The idea of disparate clustering through a relation is closely connected to the current topic of mining multiple, **alternative**, clusterings [1, 18]. Alternative clusterings are to be expected in high dimensional datasets where different explanations of the data may involve using distinct subspaces of the data. For instance, Fig. 1 (right) can be viewed as finding alternative clusterings for different types of objects (companies and countries). The clusterings of (i) the companies and (ii) the countries are alternative in the sense that we cannot use the relational information to recover one from the other and hence they are alternatives with respect to the relational information. To our knowledge, the literature on alternative clustering has not explored this scenario of alternative clustering of objects of different types.

4	0	0
0	6	0
0	0	4

2	1	1
2	1	2
1	1	3

Figure 2: Contingency tables for (left) dependent clustering and (right) disparate clustering for the scenarios from Fig. 1.

2. CLUSTERING USING CONTINGENCY TABLES: A FRAMEWORK

As stated in the introduction, we require our clusters to have two key properties. First, the individual clusters must be local in the respective attribute spaces. Second, when compared across relationships, the clusters must be either highly dependent on each other, or highly independent of each other. We present a uniform framework based on contingency tables that works for both dependent and disparate clusterings.

Fig. 2 presents contingency tables for the two clusterings from Fig. 1. The tables are 3×3, where the rows denote the clusters from the left domain (here, company clusters) and the columns denote the clusters from the right domain (here, country clusters). The cells indicate the number of entries from the corresponding clusters that are related in the original dataset. For instance, cell (1,1) of Fig. 2 (left) indicates that there are 4 relationships between entities in Cluster 1 of the companies dataset and entities in Cluster 1 of the countries dataset. Observe that the actual sizes of the clusters are not reflected in this matrix, just the number of relationships. Contrast this cell with the corresponding entry of the disparate case, which shows the smaller number of relationships (viz. 2) obtained from a different clustering.

Thus the ideal dependent case is best modeled by a diagonal or permutation contingency matrix. In practice, we can aim to achieve a diagonally dominant matrix. Similarly, the disparate case is modeled by a uniform (or near uniform) distribution over all the contingency table entries. **It is important to note, however, that we do not have direct control over the contingency table entries.** These entries are computed from the clusters, which are in turn defined by the prototype vectors. So the only variables that we can adjust are the prototype vectors but the optimization criteria must be stated in terms of the resulting contingency tables. Mathematically,

$$\begin{aligned} \text{Obj} &= \mathcal{F}(\text{Contingency table}) \\ &= \mathcal{F}(n(\text{Clustering1}, \text{Clustering2}, \text{Relation})) \\ &= \mathcal{F}(n(v(\text{Dataset1}, \text{Prototypes1}), \\ &\qquad v(\text{Dataset2}, \text{Prototypes2}), \\ &\qquad \text{Relation})) \end{aligned} \quad (1)$$

In reverse order,

- v is the clustering (assignment) function that finds (separate) clusterings of the two datasets using prototypes;

- n brings the clusterings together and prepares the contingency table w.r.t. the underlying relation;

- Finally, $\mathcal{F}$ is the objective function that evaluates the

contingency matrix for either a dependent or an disparate clustering (more on this later).

Here, the free parameters are the prototypes (Prototypes1, Prototypes2) and the objective function Obj is meant to be either minimized (for disparate clustering) or maximized (for dependent clustering).

3. FORMALISMS

Let $\mathcal{X}$ and $\mathcal{Y}$ be two datasets, where $\mathcal{X} = \{\mathbf{x}_s\}, s = 1, \ldots, n_x$ is the set of (real-valued) vectors in dataset $\mathcal{X}$, where each vector is of dimension l_x, i.e., $\mathbf{x}_s \in \mathbb{R}^{l_x}$ (Likewise $\mathcal{Y} = \{\mathbf{y}_t\}, t = 1, \ldots, n_y, \mathbf{y}_t \in \mathbb{R}^{l_y}$). The many-to-many relationships between $\mathcal{X}$ and $\mathcal{Y}$ are represented by a $n_x \times n_y$ binary matrix B, where $B(s,t) = 1$ if $\mathbf{x}_s$ is related to $\mathbf{y}_t$, else $B(s,t) = 0$. Let $C_{(x)}$ and $C_{(y)}$ be the cluster indices, i.e., indicator random variables, corresponding to the datasets $\mathcal{X}$ and $\mathcal{Y}$ and let k_x and k_y be the corresponding number of clusters. Thus, $C_{(x)}$ takes values in $\{1, \ldots, k_x\}$ and $C_{(y)}$ takes values in $\{1, \ldots, k_y\}$. We present our formalisms in accordance with the structure of Eqn. 1 so that they can then be composed to yield the objective function.

3.1 Assigning data vectors to clusters

Let $\mathbf{m}_{i,\mathcal{X}}$ be the prototype vector for cluster i in dataset $\mathcal{X}$ (similarly $\mathbf{m}_{j,\mathcal{Y}}$). (These are precisely the quantities we wish to estimate/optimize for, but in this section, assume they are given). Let $v_i^{(\mathbf{x}_s)}$ (likewise $v_j^{(\mathbf{y}_t)}$) be the cluster membership indicator variables, i.e., the probability that data sample $\mathbf{x}_s$ is assigned to cluster i in dataset $\mathcal{X}$ (resp). Thus, $\sum_{i=1}^{r^x} v_i^{(\mathbf{x}_s)} = \sum_{j=1}^{r^y} v_j^{(\mathbf{y}_t)} = 1$. The traditional k-means *hard* assignment is given by:

$$v_i^{(\mathbf{x}_s)} = \begin{cases} 1 & \text{if } ||\mathbf{x}_s - \mathbf{m}_{i,\mathcal{X}}|| \leq ||\mathbf{x}_s - \mathbf{m}_{i',\mathcal{X}}||,\ i' = 1 \ldots k_x, \\ 0 & \text{otherwise.} \end{cases}$$

(Likewise for $v_j^{(\mathbf{y}_t)}$.) Ideally, we would like a continuous function that tracks these hard assignments to a high degree of accuracy. A standard approach is to use a Gaussian kernel to smooth out the cluster assignment probabilities. Here, we present a novel smoothing formulation which provides tunable guarantees on its quality of approximation and for which the Gaussian kernel is a special case. First we define

$$\gamma_{(i,i')}(\mathbf{x}_s) = \frac{||\mathbf{x}_s - \mathbf{m}_{i',\mathcal{X}}||^2 - ||\mathbf{x}_s - \mathbf{m}_{i,\mathcal{X}}||^2}{D},\ 1 \leq i, i' \leq k_x,$$

where

$$D = \max_{s,s'} ||\mathbf{x}_s - \mathbf{x}_{s'}||^2, 1 \leq s, s' \leq n_x$$

is the pointset diameter. We now use $\operatorname{argmin}_{i'} \gamma_{(i,i')}(\mathbf{x}_s)$ for cluster assignments so the goal is to track $\min_{i'} \gamma_{(i,i')}(\mathbf{x}_s)$ with high accuracy. The approach we take is to use the Kreisselmeier-Steinhauser (KS) envelope function [13] given by

$$KS_i(\mathbf{x}_s) = \frac{-1}{\rho} \ln \Big[\sum_{i'=1}^{k_x} \exp(-\rho \gamma_{(i,i')}(\mathbf{x}_s)) \Big],$$

where $\rho \gg 0$. The KS function is a smooth function that is **infinitely differentiable** (i.e., its first, second, 3rd, .. derivatives exist). Using this the cluster membership indicators are redefined as:

$$\begin{aligned} v_i^{(\mathbf{x}_s)} &= \frac{\exp\Big[\rho KS_i(\mathbf{x}_s)\Big]}{\sum_{i'=1}^{k_x} \exp\Big[\rho KS_{i'}(\mathbf{x}_s)\Big]} \\ &= \frac{\exp(-\frac{\rho}{D}||\mathbf{x}_s - \mathbf{m}_{i,\mathcal{X}}||^2)}{\sum_{i'=1}^{k_x} \exp(-\frac{\rho}{D}||\mathbf{x}_s - \mathbf{m}_{i',\mathcal{X}}||^2)} \end{aligned} \quad (2)$$

An analogous equation holds for $v_j^{(\mathbf{y}_t)}$. The astute reader would notice that this is really the Gaussian kernel approximation with ρ/D being the width of the kernel. However, this novel derivation helps tease out how the width must be set in order to achieve a certain quality of approximation. Notice that D is completely determined by the data but ρ is a user-settable parameter, and precisely what we can tune.

3.2 Preparing contingency tables

Preparing the $k_x \times k_y$ contingency table (to capture the relationships between entries in clusters across $\mathcal{X}$ and $\mathcal{Y}$) is now straightforward. We simply iterate over every combination of data entities from $\mathcal{X}$ and $\mathcal{Y}$, determine whether they have a relationship, and suitably increment the appropriate entry in the contingency table:

$$w_{ij} = \sum_{s=1}^{n_x} \sum_{t=1}^{n_y} B(s,t) v_i^{(\mathbf{x}_s)} v_j^{(\mathbf{y}_t)}, \quad (3)$$

We also define

$$w_{i.} = \sum_{j=1}^{k_y} w_{ij}, \qquad w_{.j} = \sum_{i=1}^{k_x} w_{ij}$$

where $w_{i.}$ and $w_{.j}$ are the row-wise and column-wise counts of the cells of the contingency table respectively.

We will find it useful to define the row-wise random variables $\alpha_i, i = 1, \ldots, k_x$ and column-wise random variables $\beta_j, j = 1, \ldots, k_y$ with probability distributions as follows

$$p(\alpha_i = j) = p(C_{(y)} = j | C_{(x)} = i) = \frac{w_{ij}}{w_{i.}}, \quad (4)$$

$$p(\beta_j = i) = p(C_{(x)} = i | C_{(y)} = j) = \frac{w_{ij}}{w_{.j}}. \quad (5)$$

The row wise distributions represent the conditional distributions of the clusters in dataset in $\mathcal{X}$ given the clusters in $\mathcal{Y}$; the column wise distributions are also interpreted analogously.

3.3 Evaluating contingency tables

Now that we have a contingency table, we must evaluate it to see if it reflects a dependent or disparate set of clusterings (as the requirement may be). Ideally, we would like one criterion that when minimized leads to a disparate clustering and when maximized leads to a dependent clustering.

For this purpose, we compare the row-wise and column-wise distributions from the contingency table entries to the uniform distribution. (In the example from Fig. 2, there are three row-wise distributions and three column-wise distributions.) For dependent clusters, the row-wise and column-wise distributions must be far from uniform, whereas for disparate clusters, they must be close to uniform. We use

KL-divergences to define our unified objective function:

$$\mathcal{F} = \frac{1}{k_x}\sum_{i=1}^{k_x} D_{KL}\Big(\alpha_i||U(\frac{1}{k_y})\Big) + \frac{1}{k_y}\sum_{j=1}^{k_y} D_{KL}\Big(\beta_j||U(\frac{1}{k_x})\Big). \tag{6}$$

where the KL-divergence between distributions $p_1(x)$ and $p_2(x)$ over the sample space X is given by:

$$D_{KL}[p_1||p_2] = \sum_{x\in X} p_1(x)\log\frac{p_1(x)}{p_2(x)}$$

$D_{KL}[p_1||p_2]$ measures the inefficiency of assuming that the distribution is p_2 when the true distribution is actually p_1.

Note that the row-wise distributions take values over the columns $1,\ldots,k_y$ and the column-wise distributions take values over the rows $1,\ldots,k_x$. Hence the reference distribution for row-wise variables is over the columns, and vice versa. Also, observe that the row-wise and column-wise KL-divergences are averaged to form $\mathcal{F}$. This is to mitigate the effect of lopsided contingency tables ($k_x \gg k_y$ or $k_y \gg k_x$) wherein it is possible to optimize $\mathcal{F}$ by focusing on the "longer" dimension without really ensuring that the other dimension's projections are close to uniform.

Finally observe that the KL-divergence of any distribution with respect to the uniform distribution is proportional to the negative *entropy* ($-H$) of the distribution. Thus we are essentially aiming to minimize or maximize (for dependent or independent clusters) the entropy of the cluster conditional distributions between pairs of two datasets.

4. ALGORITHMS

Now we are ready to formally present our data mining algorithms as optimization over the space of prototypes.

4.1 Disparate clustering

Here the goal is to minimize $\mathcal{F}$, a non-linear function of $\mathbf{m}_{i,\mathcal{X}}$ and $\mathbf{m}_{j,\mathcal{Y}}$. For this purpose, we adopt an augmented Lagrangian formulation with a quasi-Newton trust region algorithm. We require a flexible formulation with equality constraints (i.e., that mean prototypes lie on the unit hypersphere) and bound constraints (i.e., that the prototypes are bounded by the max and min (componentwise) of the data, otherwise the optimization problem has no solution). The LANCELOT software package [6] provides just such an implementation.

For ease of description, we "package" all the mean prototype vectors for clusters from both datasets (there are k_x+k_y of them) into a single vector ν of length t. The problem to solve is then:

$$\text{argmin}\,\mathcal{F}(\nu) \qquad \text{subject to} \qquad h_i(\nu) = 0,\ i = 1,\ldots,\eta,$$
$$L_j \le \nu_j \le U_j, \quad j = 1,\ldots,t.$$

where ν is a t-dimensional vector and $\mathcal{F}$, h_i are real-valued functions continuously differentiable in a neighborhood of the box $[L,U]$. Here the h_i ensure that the mean prototypes lie on the unit hypersphere (i.e., they are of the form $||\mathbf{m}_{1,\mathcal{X}}||-1,\ ||\mathbf{m}_{2,\mathcal{X}}||-1,\ \cdots,\ ||\mathbf{m}_{1,\mathcal{Y}}||-1,\ ||\mathbf{m}_{2,\mathcal{Y}}||-1,\ \cdots$) The bound constraints are uniformly set to $[-1,1]$. The augmented Lagrangian Φ is defined by

$$\Phi(\nu,\lambda,\varphi) = \mathcal{F}(\nu) + \sum_{i=1}^{\eta}\big(\lambda_i h_i(\nu) + \varphi h_i(\nu)^2\big), \tag{7}$$

where the λ_i are Lagrange multipliers and $\varphi > 0$ is a penalty parameter. The augmented Lagrangian method (implemented in LANCELOT) to solve the constrained optimization problem above is given in *OptPrototypes*.

Algorithm 1 *OptPrototypes*

1. Choose initial values $\nu_{(0)}$ (e.g., via a k-means algorithm), $\lambda_{(0)}$, set $k := 0$, and fix $\varphi > 0$.
2. For fixed $\lambda_{(k)}$, update $\nu_{(k)}$ to $\nu_{(k+1)}$ by using one step of a quasi-Newton trust region algorithm for minimizing $\Phi\big(\nu,\lambda_{(k)},\varphi\big)$ subject to the constraints on ν. Call *ProblemSetup* with ν as needed to obtain $\mathcal{F}$ and $\nabla\mathcal{F}$.
3. Update λ by $\lambda_{(k+1)_i} = \lambda_{(k)_i} + 2\varphi h_i\big(\nu_{(k)}\big)$ for $i = 1,\ldots,\eta$.
4. If $\big(\nu_{(k)},\lambda_{(k)}\big)$ has converged, stop; else, set $k := k+1$ and go to (2).
5. Return ν.

In Step 1 of *OptPrototypes*, we initialize the prototypes using a k-means algorithm (i.e., one which separately finds clusters in each dataset without coordination), package them into the vector ν, and use this vector as starting points for optimization. For each iteration of the augmented Lagrangian method, we require access to $\mathcal{F}$ and $\nabla\mathcal{F}$ which we obtain by invoking Algorithm *ProblemSetup*.

Algorithm 2 *ProblemSetup*

1. Unpackage ν into values for mean prototype vectors.
2. Use Eq. (2) (and its analog) to compute $v_i^{(\mathbf{x}_s)}$ and $v_j^{(\mathbf{y}_t)}$.
3. Use Eq. (3) to obtain contingency table counts w_{ij}.
4. Use Eqs. (4) and (5) to define r.v.s α_i and β_j.
5. Use Eqn. (6) to compute $\mathcal{F}$ and $\nabla\mathcal{F}$ (see [19].)
6. Return $\mathcal{F}, \nabla\mathcal{F}$.

This routine goes step-by-step through the framework developed in earlier sections to link the prototypes to the objective function. There are no parameters in these stages except for ρ which controls the accuracy of the KS approximations. It is chosen so that the KS approximation error is commensurate with the optimization convergence tolerance. Gradients (needed by the trust region algorithm) are mathematically straightforward but tedious, so are not explicitly given here (see [19]).

Modulo the time complexity of k-means (which is used for initializing the prototypes), the per-iteration complexity of the various stages of our algorithm can be given as follows:

Step	Time Complexity
Assigning vectors to clusters	$\mathcal{O}(n_x l_x k_x + n_y l_y k_y)$
Preparing contingency tables	$\mathcal{O}(k_x k_y n_x n_y)$ (naïve)
	$\mathcal{O}(k_x k_y \lvert\mathcal{B}\rvert)$ (replicated)
Evaluating contingency tables	$\mathcal{O}(k_y k_x + k_x k_y)$
Optimization	$\mathcal{O}((\eta+1)t^2)$

First, observe that this is a continuous, rather than discrete, optimization algorithm, and hence the overall time complexity depends on the number of iterations, which is an unknown function of the requested numerical accuracy. The step of assigning vectors to clusters takes place independently in the two datasets, so the time complexity has two components. For each vector, we compare it to each mean prototype, and an inner loop over the dimensionality

1	0	0
0	1	0
0	0	14

1.5	1.5	1.5
1.5	1.5	1.5
1.5	1.5	1.5

Figure 3: Degenerate contingency tables for (left) dependent clusters and (right) disparate clusters. These are bad solutions to be avoided because the clusters in (a) are highly imbalanced and (b) is obtained by trivially assigning all points to all clusters.

of the vectors gives $\mathcal{O}(n_x l_x k_x + n_y l_y k_y)$. The straightforward way to prepare contingency tables as suggested by Eqn. 3 gives rise to a costly computation, since for each cell of the contingency table (there are $k_x k_y$ of them), we will expend $\mathcal{O}(n_x n_y)$ computations. In [19] we show how we can reduce this by an order of magnitude using a method of 'replicating' vectors which helps us treat the relationship matrix $\mathcal{B}$ as if it were one-to-one. In this case, the per-cell complexity will be simply be a linear function of the non-zero entries in $\mathcal{B}$, i.e., $|\mathcal{B}|$. Evaluating the contingency tables requires us to calculate KL-divergences which are dependent on the sample space over which the distributions are compared and the number of such comparisons. There are two terms, one for row-wise distributions, and one for column-wise distributions. Finally, the time complexity of the optimization is $\mathcal{O}((\eta+1)t^2)$ per iteration, and the space complexity is also $\mathcal{O}((\eta+1)t^2)$, mostly for storage of Hessian matrix approximations of $\mathcal{F}$ and h_i. Note that $t = k_x l_x + k_y l_y$ and $\eta = k_x + k_y$. In practice, to avoid sensitivity to local minima, we perform several random restarts of our approach, with different initializations of the prototypes.

Dependent clustering proceeds exactly as above except the goal now is to min $-\mathcal{F}$ (i.e., to maximize $\mathcal{F}$). Simply replacing $\mathcal{F}$ with $-\mathcal{F}$ in the above algorithm conducts dependent clustering. For ease of description later, we henceforth refer to $\mathcal{F}$ as $\mathcal{F}_{\text{indep}}$ and to $-\mathcal{F}$ as $\mathcal{F}_{\text{dep}}$.

4.2 Regularization

Degenerate situations can arise as shown in Fig. 3. In the dependent case, we might obtain a diagonal contingency table but with imbalanced cluster sizes. In the independent case, the data points are assigned with equal probability to every cluster, resulting in a trivial solution for ensuring that the contingency table resembles a uniform distribution. See [19] for how to add additional terms in the objective function to alleviate both these issues.

A final issue is the determination of the right number of clusters, which has a direct mapping to the sufficient statistics of contingency tables necessary to capture differences between distributions. We have used the minimum discrimination information (MDI) principle (discussed later) for model selection. Due to space limitations, we are unable to cover this aspect in detail.

5. EXPERIMENTS

We evaluate our approach using both synthetic and real datasets. The questions we seek to answer through our experiments are:

1. Can we realize classical constrained clustering and alternative clustering scenarios (i.e., over a single dataset) using our framework? (Sections 5.1 and 5.2)

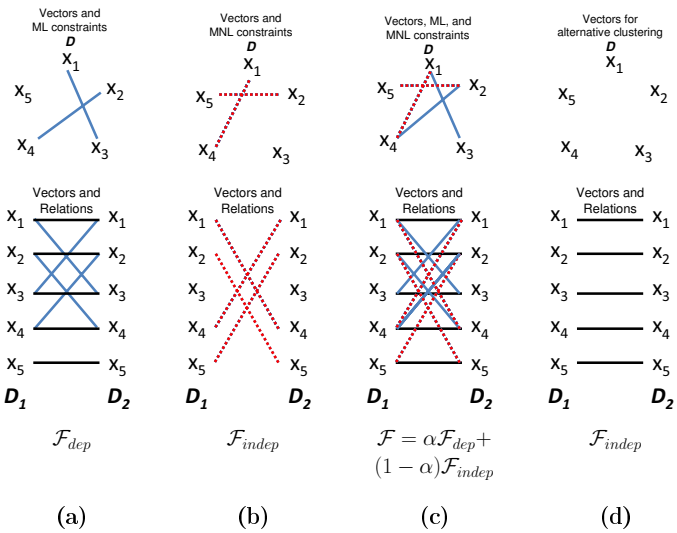

Figure 4: Realizing classical single-dataset clustering scenarios using our framework. (a) Clustering with must-link constraints. (b) Clustering with must-not-link constraints. (c) Clustering with both must-link and must-not-link constraints. (d) Finding alternative clusterings.

2. How much does our emphasis on clustering relations compromise locality of clusters in the respective attribute spaces? (Section 5.3)

3. How does our approach (of defining an integrated objective function and locally minimizing it) scale? (Section 5.3)

4. As the number of clusters increases, does it become easier or more difficult to achieve dependent and disparate clusterings? (Section 5.3)

5. Can we pose integrated dependent and disparate clustering formulations over non-homogeneous data involving multiple datasets and relations? (Section 5.4)

6. In mining non-homogeneous datasets with multiple criteria, what is the effect of varying the emphasis of different criteria on the clustering results? (Section 5.5)

5.1 Constrained Clustering

In constrained clustering, we are given a single dataset $\mathcal{D}$ with instance level constraints such as must-link and must-not-link constraints [7, 20]. We can model such problems in our relational context as shown in Fig. 4 (a), (b), and (c). We create two copies of $\mathcal{D}$ into $\mathcal{D}_1$ and $\mathcal{D}_2$. In the case with only must-link (ML) constraints (Fig. 4 (a)), such as between x_1 and x_3, we create a relation between the entries: x_1 of $\mathcal{D}_1$ and x_3 of $\mathcal{D}_2$, and between entries: x_3 of $\mathcal{D}_1$ and x_1 of $\mathcal{D}_2$. In addition we include relations between the **same instances** in $\mathcal{D}_1$ and $\mathcal{D}_2$. Applying the dependent clustering criterion $\mathcal{F}_{\text{dep}}$ on this dataset will realize the constrained clustering scenario. Conversely, as shown in Fig. 4 (b), for must-not-link (MNL) constraints we would create relations between the entries that should not be brought together, and use $\mathcal{F}_{\text{indep}}$ as the optimization criterion. In Fig. 4 (a), the relations would force the clusterings to be dependent and as

Figure 5: Comparison of our approach with unconstrained k-means and two other constrained clustering formulations. We cluster the Iris dataset with randomly generated 100 ML and 100 MNL constraints. Results are averaged over 20 runs each. (left) Number of constraints violated. (middle) Objective function. (right) Normalized mutual information.

a result, either clustering would respect the ML constraints. In Fig. 4 (b), the $\mathcal{F}_{\text{indep}}$ objective will force the clusterings to violate the relations (which are really MNL constraints).

Going further, we can combine the above modeling approaches in Fig. 4 (c) which has both ML and MNL constraints. For this scenario, the optimization criterion is essentially a convex combination of both $\mathcal{F}_{\text{dep}}$ and $\mathcal{F}_{\text{indep}}$. As we vary α smoothly from 0 to 1, we increase our emphasis from satisfying the ML constraints to satisfying the MNL constraints. Here we set α to 0.5 (and explore other settings in future sections). We compare our constrained clustering framework with simple unconstrained k-means and two constrained k-means algorihtms (MPCK-MEANS and PCK-MEANS) from [4]. In overall, the number of constraint violations from our approach (Fig. 5 (left)) is worse than that of either MPCK-MEANS and PCK-MEANS, except for a small number of clusters. This is to be expected since our method does not take a strict (boolean) view of constraint satisfactions. Conversely, the objective function in our approach is the best possible value (Fig. 5 (middle)) when compared with the solutions obtained by the other three algorithms. Finally, as shown in Fig. 5 (right), the normalized mutual information score (between the cluster assignments and the class labels) is best for our approach than for the other three algorithms. This shows that taking a soft view of constraints does not compromise the locality of the mined clusters.

5.2 Finding Alternative Clusterings

We investigate alternative clustering using the Portait dataset as studied in [11]. This dataset comprises 324 images of three people each in three different poses and 36 illuminations. Pre-processing involves dimensionality reduction to a grid of 64×49 pixels. The goal of finding two alternative clusterings is to assess whether the natural clustering of the images (by person and by pose) can be recovered. We utilize the same 300 features as used in [11] and setup our framework as shown in Fig. 4 (d). Two copies of the dataset are created with one-to-one relationships and we aim to cluster the dataset in a disparate manner.

Table 1: Contingency tables in analysis of the Portrait dataset. (a) After k-means with random initializations. (b) after disparate clustering.

(a)

	C_1	C_2	C_3
C_1	0	0	72
C_2	63	64	0
C_3	3	8	114

(b)

	C_1	C_2	C_3
C_1	36	36	36
C_2	36	36	36
C_3	36	36	36

Table 2: Accuracy on the Portrait dataset.

Method	Person	Pose
k-means	0.65	0.55
Conv-EM [11]	0.69	0.72
Dec-kmeans [11]	0.84	0.78
Our framework (disparate)	**0.93**	**0.79**

Table 1 shows the two contingency tables in the analysis of the Portrait dataset and table 2 depicts the achieved accuracies using simple k-means, convolutional-EM [11], decorrelated-kmeans [11] and our framework for disparate clustering. Our algorithm performs better than all other tested algorithms according to both person and pose clusterings.

Fig. 6 shows how the accuracies of the person and the

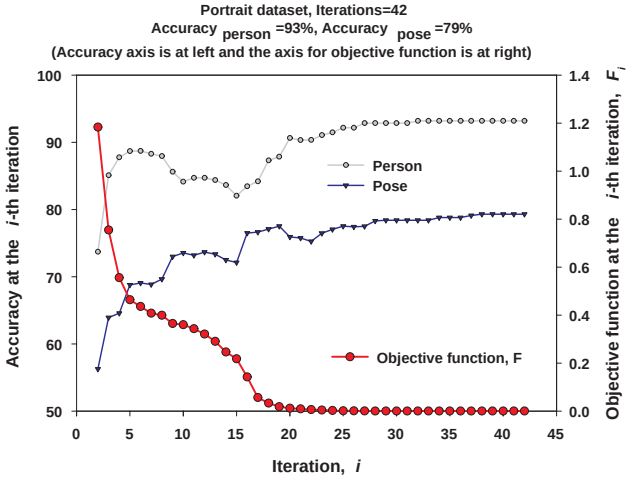

Figure 6: Monotonic improvement of objective function (finding alternative clusterings for the Portrait dataset).

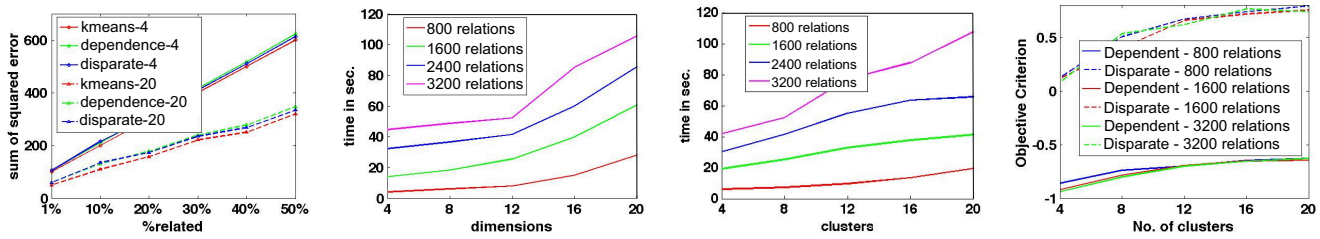

Figure 7: Synthetic data results. (a) Comparisons of SSE measures with k-means. (b&c) Time as number of attribute dimensions (b) or as number of clusters (c) is increased. (d) Objective criterion as a function of the number of clusters for both dependent and disparate schemas of clustering.

pose clusterings improve over the iterations, as the objective function is being minimized. The quasi-Newton trust region algorithm guarantees the monotonic improvement of the objective function without directly enforcing error metrics over the feature space. But because the objective function captures the dissimilarity between the two clusterings, indirectly, we notice that the accuracies w.r.t. the two alternative clusterings improve with the increase in number of iterations (though, not monotically).

5.3 Scalability and Locality Preservation

In this section, we consider two synthetic datasets with one (possibly many-many) relationship between them. The parameters we study are: l_x, l_y, the dimensions of the vectors (varied from 4 to 20); n_x, n_y, the number of vectors (fixed at 100, because as our time complexity analysis shows, they only affect the step of assigning vectors to clusters); k_x, k_y, the number of clusters sought (also varied from 4 to 20; and $|\mathcal{B}|$, the number of relationships between the datasets (varied from a one-to-one case to about a density of 50%). The vectors were themselves sampled from (two different) mixture-of-Gaussians models.

Fig. 7 (a) answers the question of whether our approach yields local clusters as the number of relationships increase (and hence each dataset is more influenced by the other). In this figure, we used settings of 4 and 20 clusters and used our framework to find dependent as well as disparate clusters, and also compared them with k-means (which doesn't use the relationship). Fig. 7 (a) shows that even though the k-means algorithm is mining two separate datasets independently, our algorithms achieve very closely comparable results in spite of the co-ordination (dependence or disparate) requirements. Thus, locality of clusters in their respective attribute spaces is not compromised and unvarying with the sparsity of the relationship matrix. At the same time, as Fig. 8 shows (for the case of four clusters), we achieve the specified contingency table criteria.

7	6	4	11
1	3	10	2
9	10	3	8
4	9	2	11

17	3	2	2
4	18	2	3
4	3	20	3
2	0	1	16

6	5	7	5
8	7	5	5
6	6	7	4
7	8	7	7

Figure 8: Our approach helps drive a k-means cluster assignment (top) toward either dependent (bottom left) or disparate (bottom right) sets of clusters.

Fig. 7 ((b),(c)) shows the runtime for our algorithm as a function of attribute vector dimensions (i.e., l_x, l_y) and number of clusters (i.e., k_x, k_y). We vary one parameter, keeping the other fixed (k_x, k_y fixed at 8 versus l_x, l_y fixed at 12). In overall these plots track the complexity analysis presented earlier except for the higher dimension/cluster settings which show steeper increases in time. This can be attributed to the greater number of iterations necessary for convergence in these cases.

Finally, we explore how our results are influenced by the number of clusters, for both dependent as well as disparate clustering formulations. (see Fig. 7 (d)). As the number of clusters increases, both objective criteria ($\mathcal{F}_{\text{dep}}$ and $\mathcal{F}_{\text{indep}}$) become difficult to attain, but for different reasons (recall that the intent of both criteria is to be minimized). In the case of dependent clusters, although the problem gets easier as clusters increase (every point can become its own cluster), the objective function scores get lower due to our regularization as explained in Section 4. In the case of disparate clusters, as the number of clusters increases, the size of the contingency table increases quadratically with the number of samples staying constant. As a result, it becomes difficult to distribute the samples across the contingency table entries without introducing some level of dependence (i.e., some entries must be zero implying dependence).

5.4 Comparing gene expression programs across yeast, worm, and human

In this study, we focus on time series gene expression profiles collected over heat shock experiments done on organisms of varying complexity: $\mathcal{H}$: human cells (4 time points) [17], $\mathcal{Y}$: yeast (8 time points) [10], and $\mathcal{C}$: *C. elegans* (worm; 7 time points) [16]. We also gathered many-many (top-k) ortholog information between the three species. A typical goal in multi-species modeling is to identify both conserved gene expression programs as well as differentiated gene expression programs. The former is useful for studying core metabolism and stress response processes, whereas the latter is useful for studying species-specific functions (e.g., the yeast is more tolerant to desiccation stress, but the worm is the more complex eukaryote).

First we study a 3-way clustering setup with only two constraints, namely that clusters in $\mathcal{H}$ and $\mathcal{W}$ must be de-

pendent, denoted by $\mathcal{H} = \mathcal{W}$, and that clusters in $\mathcal{W}$ and $\mathcal{Y}$ must be disparate, denoted by $\mathcal{W} <> \mathcal{Y}$ (See Fig. 9 (top)). As the balance between these criteria is varied from one extreme to another (via the convex combination formulation), this curve traces out the objective function values. The top left corner is the point where complete emphasis is placed on achieving the $\mathcal{H} = \mathcal{W}$ criterion (conversely for the bottom right corner). As we seek to improve the other criteria, note that we might (and will) sacrifice the already achieved criterion. The point of maximum curvature on this plot gives a 'sweet spot' so that any movement away from the sweet spot would cause a dramatic change in the objective function values. A qualitatively different type of plot is shown in Fig. 9 (middle) (for the case study described in the next section) but here again the point of maximum curvature reveals a balancing threshold of the two criteria. A 3-way clustering setup with three constraints is described in Fig. 10 and its corresponding tradeoff plot is in Fig. 9 (bottom). Here there are likely multiple points of interest depending on which criteria are sacrificed in favor of others.

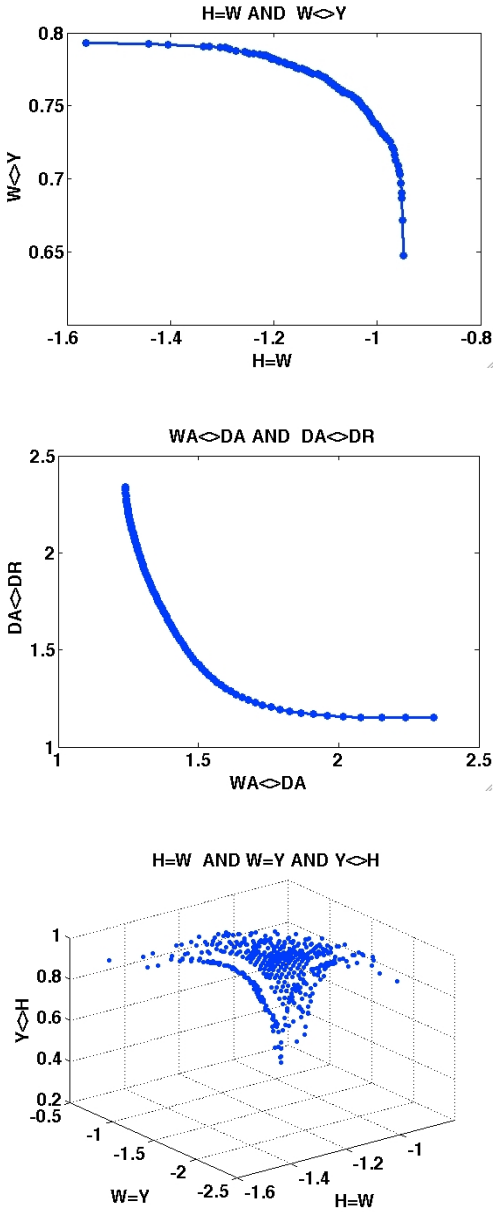

Figure 9: Balancing objectives in multi-criteria clustering optimization. Points of maximum curvature on these plots reveals a balancing point between the conflicting criteria.

5.5 Multi-organismal and multi-stress modeling

Finally, we present a case study that has a diversity of both organisms and stresses. To capture process-level similarities and differences, the data vectors we cluster here correspond to Gene Ontology categories rather than individual gene expression profiles. We used three time series datasets: CA–*C.elegans* aging (7 time points), DA–*D. melanogaster* aging (7 time points) and DR–*D. melanogaster* caloric restriction (9 time pints). Observe that the first two datasets share a similarity of process whereas the latter two share a similarity of organism. In a sense, the *D. melanogaster* aging dataset is squarely in the "middle" of the other two datasets. When subjected to clustering together, the inherent tradeoff is what we seek to capture.

For this evaluation, we studied the enrichment of clusters obtained from our framework vis-a-vis k-means clustering. We set the number of clusters at 7 and evaluated GO terms for an FDR-corrected q-value of 0.05. First, we study the clustering setup so that DA=DR AND CA=DA, for a setting of $\alpha = 0$ (so that more emphasis is placed on achieving the dependent clustering DA=DR). Here, we observed 75 GO terms enriched (versus 37 for k-means). Similar improvements were seen for $\alpha = 0.5$ (55 versus 20) and for $\alpha = 1$ (89 versus 35). Observe the greater numbers of terms enriched in general for the extremalities (which is to be expected). In terms of process-level similarities, the GO terms common across the aging datasets but which do not appear when we emphasize organism-level similarities are:

> neuron recognition, embryonic pattern specification, aromatic compound catabolic process, somatic sex determination, sulfur compound biosynthetic process.

Conversely, the organism-level similarities are captured in:

> chemosensory behavior, peptide metabolic process, regulation of cell proliferation, anatomical structure formation, cell redox homeostasis, negative regulation of growth.

These results show that process-level similarities involve higher order functions whereas organism-level similarities involve growth and metabolism processes. The careful interplay between aging and caloric restriction, both at the organismal and at the inter-organismal level, is an interesting conclusion from this study.

6. RELATED WORK

MDI: The objective functions defined here have connections to the principle of minimum discrimination information (MDI), introduced by Kullback for the analysis of contingency tables [14] (the minimum Bregman information

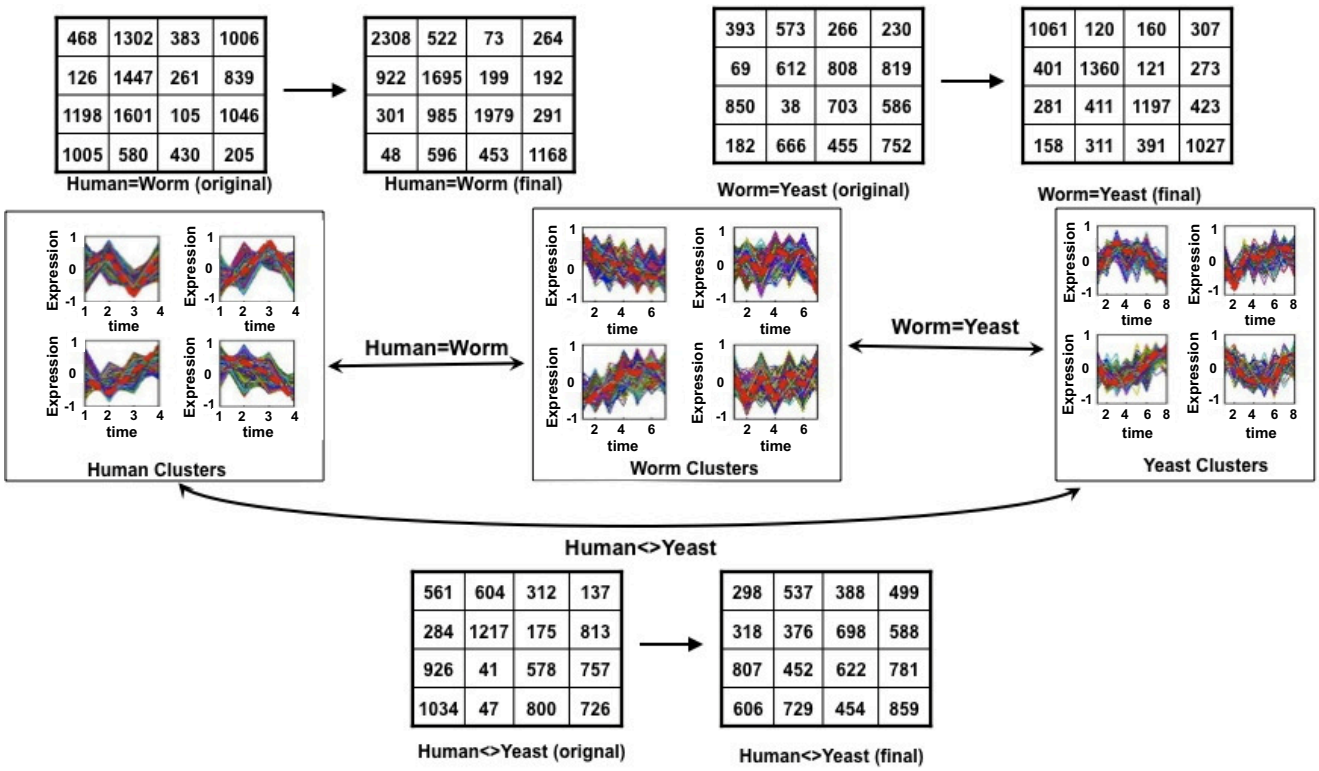

Figure 10: Clustering three datasets with three constraints between them. Two sets of clusters (between human/worm and between worm/yeast) are constrained to be similar whereas the third set (between human/yeast) is constrained to be dissimilar. Observe how the top two contingency tables are driven toward diagonal dominance whereas the bottom contingency table is driven toward a uniform distribution.

(MBI) in [3] can be seen as a generalization of this principle). The MDI principle states that if q is the assumed or true distribution, the estimated distribution p must be chosen such that $D_{KL}(p||q)$ is minimized. In our objective functions the estimated distribution p is obtained from the contingency table counts. The true distribution q is always assumed to be the uniform distribution. We maximize or minimze the KL-divergence from this true distribution as required. Space restrictions prevent us from describing the connection to MDI in further detail.

Co-clustering binary matrices, Associative clustering, and Cross-associations: Identifying clusterings over a relation (i.e., a binary matrix) is the topic of many efforts [5, 8]. The former uses information-theoretic criteria to best approximate a joint distribution of two binary variables and the latter uses the MDL (minimum description length) principle to obtain a parameter-less algorithm by automatically determining the number of clusters. Our work is focused on not just binary relations but also attribute-valued vectors. The idea of comparing clustering results using contingency tables was first done in [12] although our work is the first to unify dependent and disparate clusterings in the same framework.

Finding disparate clusterings: The idea of finding disparate clusterings has been studied in [11]. Here only one dataset is considered and two **dissimilar** clusterings are sought **simultaneously** where the definition of dissimilarity is in terms of orthogonality of the two sets of basis vectors. This is an indirect way to capture dissimilarity whereas in our paper we use contingency tables to more directly capture the dissimilarity. Furthermore, our work enables the combination of similar clusterings and disparate clusterings in a more expressive way. For instance, given just two datasets X and Y with two relationships R1 and R2 between them, our work can identify clusters in X and Y that are similar from the perspective of R1 but dissimilar from the perspective of R2: it is diffcult to specify such criteria in terms of the basis vectors since they will be the same irrespective of the relationship.

Clustering over relation graphs: Clustering over relation graphs is a framework by Banerjee et al. [2] that uses the notion of Bregman divergences to unify a variety of loss functions and applies the Bregman information principle (from [3]) to preserve various summary statistics defined over parts of the relational schema. This framework can handle all the types of data and relationships we study here, since the notion of Bregman divergences is very general and can capture both information-theoretic criteria (from our contingency tables) as well as geometric measures (for our locality of clusters). However, our work unifies dependent with disparate clustering, whereas Banerjee et al. focuses on only the dependent case. This entails several key differences. First, Banerjee et al. aim to **minimize the distortion** as defined through conditional expectations over the original random variables, whereas our work is meant to

both minimize and introduce distortion as needed, over different parts of the schema as appropriate. This leads to tradeoffs across the schema which is unlike the tradeoffs experienced in [2, 3] between compression and accuracy of modeling. (see also MIB, discussed below). A second difference is that our work does not directly minimize error metrics over the attribute-value space and uses contingency tables (relationships between clusters) to exclusively drive the optimization. This leads to the third difference, namely that the distortions we seek to minimize/maximize are w.r.t. idealized contingency tables rather than w.r.t. the original relational data. The net result of these variations is that relations (idealized as well as real) are given primary importance in influencing the clusterings.

Multivariate information bottleneck: Our work is reminiscent of the multivariate information bottleneck (MIB) [9] which is a framework for specifying clusterings in terms of two conflicting criteria: compression (of vectors into clusters) and preservation of mutual information (of clusters with auxiliary variables that are related to the original vectors). We share with MIB the formulation of a multi-criteria objective function derived from a clustering schema but differ in the specifics of both the intent of the objective function and how the clustering is driven based on the objective function. Furthermore, the MIB framework was originally defined for discrete settings whereas we support a mixed modality of datasets.

7. DISCUSSION

We have presented a very general and expressive framework for clustering non-homogeneous datasets. We have also shown how it subsumes many previously defined formulations and that it sheds useful insights into tradeoffs underlying complex relationships between datasets.

Our directions for future work are two fold. Thus far, we have used distinct relations to enforce disparate and dependent clusterings. One of the first directions for future work is to allow both types of clusterings to be captured in the same relation. This would help capture more expressive relationships between datasets, such as a banded diagonal structure in the contingency table. Secondly, just as the theory of functional and multi-valued dependencies (FDs and MDs) helps model relations in and between individual tuples, we aim to develop a theory of 'clustering dependencies' that can help model relations in the aggregate, e.g., between clusters.

8. ACKNOWLEDGMENTS

This work is supported in part by US National Science Foundation grants CCF-0937133, CNS-0615181, and the Institute for Critical Technology and Applied Science (ICTAS) at Virginia Tech.

9. REFERENCES

[1] E. Bae and J. Bailey. COALA: A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity. In *ICDM '06*, pages 53–62, 2006.

[2] A. Banerjee, S. Basu, and S. Merugu. Multi-way Clustering on Relation Graphs. In *SDM '07*, pages 225–334, 2007.

[3] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman Divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[4] M. Bilenko, S. Basu, and R. J. Mooney. Integrating Constraints and Metric Learning in Semi-supervised Clustering. In *ICML '04*, pages 11–18, 2004.

[5] D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos. Fully Automatic Cross-associations. In *KDD '04*, pages 79–88, 2004.

[6] A. R. Conn, N. I. M. Gould, and P. L. Toint. *LANCELOT: A Fortran Package for Large-scale Nonlinear Optimization (Release A)*, volume 17. Springer Verlag, 1992.

[7] I. Davidson and S. S. Ravi. Clustering with Constraints: Feasibility Issues and the k-Means Algorithm. In *SDM '05*, pages 201–211, 2005.

[8] I. S. Dhillon, S. Mallela, and D. S. Modha. Information Theoretic Co-clustering. In *KDD '03*, pages 89–98, 2003.

[9] N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate Information Bottleneck. In *UAI '01*, pages 152–161, 2001.

[10] A. P. Gasch, P.T. Spellman, C.M. Kao, Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, and P.O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11(12):4241–57, 2000.

[11] P. Jain, R. Meka, and I. S. Dhillon. Simultaneous Unsupervised Learning of Disparate Clusterings. In *SDM '08*, pages 858–869, 2008.

[12] S. Kaski, J. Nikkilä, J. Sinkkonen, L. Lahti, J.E.A. Knuuttila, and C. Roos. Associative Clustering for Exploring Dependencies between Functional Genomics Data Sets. *IEEE/ACM TCBB*, 2(3):203–216, 2005.

[13] G. Kreisselmeier and R. Steinhauser. Systematic Control Design by Optimizing a Vector Performance Index. In *IFAC Symp. on Computer Aided Design of Control Systems*, pages 113–117, 1979.

[14] S. Kullback and D.V. Gokhale. *The Information in Contingency Tables*. Marcel Dekker Inc., 1978.

[15] B. Long, X. Wu, Z. Zhang, and P. S. Yu. Unsupervised Learning on k-partite Graphs. In *KDD '06*, pages 317–326, 2006.

[16] S. A. McCarroll, C. T. Murphy, S. Zou, S. D. Pletcher, C. Chin, Y. N. Jan, C. Kenyon, C. I. Bargmann, and H. Li. Comparing genomic expression patterns across species identifies shared transcriptional profile in aging. *Nature Genetics*, 36(2):197–204, 2004.

[17] T. J. Page, D. Sikder, L. Yang, L. Pluta, R. D. Wolfinger, T. Kodadek, and R. S. Thomas. Genome-wide analysis of human hsf1 signaling reveals a transcriptional program linked to cellular adaptation and survival. *Molecular Biosystems*, 2:627–639, 2006.

[18] Z. Qi and I Davidson. A Principled and Flexible Framework for Finding Alternative Clusterings. In *KDD '09*, pages 717–726, 2009.

[19] S. Tadepalli. *Schemas of Clustering*. PhD thesis, Virginia Tech, Feb 2009.

[20] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained K-means Clustering with Background Knowledge. In *ICML '01*, pages 577–584, 2001.