On Finding Large Conjunctive Clusters

Nina Mishra[1], Dana Ron[2], and Ram Swaminathan[3]

[1] HP Labs and Stanford University.*** nmishra@theory.stanford.edu.
[2] Tel-Aviv University.† danar@eng.tau.ac.il.
[3] HP Labs. swaram@hpl.hp.com.

Abstract. We propose a new formulation of the clustering problem that differs from previous work in several aspects. First, the goal is to explicitly output a collection of simple and meaningful *conjunctive descriptions* of the clusters. Second, the clusters might overlap, i.e., a point can belong to multiple clusters. Third, the clusters might not cover all points, i.e., not every point is clustered. Finally, we allow a point to be assigned to a conjunctive cluster description even if it does not completely satisfy all of the attributes, but rather only satisfies most.
A convenient way to view our clustering problem is that of finding a collection of large bicliques in a bipartite graph. Identifying one largest conjunctive cluster is equivalent to finding a maximum edge biclique. Since this problem is NP-hard [28] and there is evidence that it is difficult to approximate [12], we solve a relaxed version where the objective is to find a large subgraph that is close to being a biclique. We give a randomized algorithm that finds a relaxed biclique with almost as many edges as the maximum biclique. We then extend this algorithm to identify a good collection of large relaxed bicliques. A key property of these algorithms is that their running time is *independent* of the number of data points and linear in the number of attributes.

Keywords: Conceptual Clustering, Max Edge Biclique, Unsupervised Learning

1 Introduction

It has become evident that clustering is not a single problem, but rather a collection of application-specific optimization problems. Conductance-based clustering [22] is appealing when the objective is to maximize edge connectivity within a cluster and minimize the weight of inter-cluster edges. Correlation clustering [6] is practical when each pair of data points can be given a label according to whether they should or should not belong to the same cluster and when the objective is to minimize the number of disagreements/maximize the number of agreements with the given labels. As clustering becomes more clearly defined for other applications, many new and important formulations will follow.

*** Research partially supported by NSF grant EIA-0137761.
† Part of this reserach was done while visiting HP Labs. The Research is supported by the Israel Science Foundation (grant number 32/00-1).

In this paper we are motivated by applications where the goal is to identify tight descriptions of large groups of points. One example is customer segmentation where one may be interested in describing common customers via a simple conjunctive description like people who care about "Price AND Quality" when making product purchase decisions. Such cluster descriptions can then be used as a basis for target marketing.

We introduce a new kind of clustering that we call *Conjunctive Clustering (CC)* where the objective is to identify long conjunctive cluster descriptions that cover a dense region of the space. More formally, a conjunctive cluster is a conjunction of attributes c together with the points Y in the data set that satisfy the conjunction c. In general we are interested in longer, more specific conjunctions since then the points that satisfy the conjunction have more in common. We are also interested in having a large number of points satisfy that conjunction. A natural way to combine these objectives is to maximize $|c| \cdot |Y|$ so that we cover as many points as possible that have much in common.

A convenient way to think about a conjunctive cluster is as a biclique in a bipartite graph. For a bipartite graph $G = (U, W, E)$, let U be the points to be clustered, W the attributes that describe the data, and let there be an edge between (u, w) if the point u has attribute w. (In the more general categorical case, there is a vertex in W for each attribute/value combination.) A biclique is a subgraph (U^*, W^*) with $|U^*| \cdot |W^*|$ edges, in other words there is an edge between each vertex in U^* and each vertex in W^*. A biclique naturally corresponds to a conjunctive cluster since each point u in U^* satisfies the conjunction of attributes in W^*. A maximum edge biclique corresponds to the best conjunctive cluster, since $|W^*|$ is precisely the length of the conjunction and $|U^*|$ is the number of points that satisfy the conjunction. We define the k best conjunctive clusters as the k largest clusters that don't overlap too much (a formal definition can be found in Section 2).

1.1 Advantages of the Conjunctive Clustering Formulation

Our interpretation of clustering as the problem of identifying large bicliques is appealing for many reasons:

Cluster Descriptions: For the applications we have in mind, it is not sufficient for a clustering algorithm to output subsets of points that each belong to the same cluster. Rather, the identification of a cluster description is crucial.

In practice, the problem of identifying cluster descriptions is overcome by using a machine learning algorithm. Points in the same cluster are assigned the same class label, and a machine learning algorithm is used to learn a function that distinguishes the classes from each other. The problem with this approach is that typically the clustering algorithm optimizes a cost function unrelated to the type of cluster description sought. Thus a common byproduct of the machine learning step is a collection of descriptions that are not necessarily conjunctive, and usually hard to understand. If on the other hand the learning algorithm is forced to output conjunctions as hypotheses, then they may serve

as poor descriptions of clusters (since the clusters found may be inherently more complex). Thus by performing clustering and learning separately, one may be sacrificing the descriptive quality of the final clusters – either because they are too complicated to understand, or they are too simple to describe the clusters.

Clusters Overlap / Some Points Not Clustered: The objective of many existing clustering algorithms is to identify a strict partition of the points [21, 8, 11, 18, 22]. In practice such a condition is far too stringent: Each point need not be clustered and further some points can be assigned to multiple clusters. By viewing a clustering as a collection of bicliques, we allow clusters to overlap, both in that each point can belong to multiple clusters and that an attribute can be used to describe multiple clusters.

We also allow our algorithm to ignore some points if they don't fall into one of the k desired clusters. In contrast to outlier detection methods, which typically are viewed as a preprocessing step employed prior to clustering, we directly allow the clustering algorithm to ignore some points. Furthermore, while clustering algorithms like EM [10] allow points to be assigned to multiple clusters, such approaches do not typically produce cluster descriptions.

Move away from Metric Space: Much of the existing research on clustering assumes that the data to be clustered falls in some metric space, e.g., k–Median [21, 8, 5, 27] and k–Center [20, 11, 18]. Such a measure is useful in situations where the goal is to ensure that pairs of points in the same cluster are "close" according to the metric space. In the CC formulation, if the points to be clustered are in $\{0,1\}^d$, for example, then it is possible that two points in the same cluster are "far" according to the metric space, but placed in the same cluster since they share some common subset of variables. In other words, some dimensions are completely ignored when placing points in the same conjunctive cluster.

Furthermore, there may be applications where data does not inherently fall in a metric space, e.g., text and images. In such situations, it may be difficult to both quantify the distance between two objects as well as ensure that the distance measure satisfies the triangle inequality. The CC view of the clustering problem moves away from quantifying the distance between points. Instead it attempts to find large subsets of points that have many attributes in common.

1.2 Our Results

Maximum Conjunctive Cluster/Maximum Edge Biclique. We start by considering the problem of finding a maximum conjunctive cluster, that is a cluster/biclique (U^*, W^*) with the most edges. Since this problem is NP hard [28] and there is evidence that it is difficult to approximate [12], we consider a relaxation of the maximum edge biclique problem where the algorithm is allowed to output a pair $(\hat{U}, \hat{W})$, where $\hat{U} \subseteq U$ and $\hat{W} \subseteq W$, that is ϵ-close to being a conjunctive cluster/biclique. That is, every point in $\hat{U}$ has at least $(1-\epsilon)$ of the attributes in $\hat{W}$. More precisely, the algorithm only outputs $\hat{W}$ (which corresponds to the cluster description), and $\hat{U}$ is implicitly determined by $\hat{W}$ (that is, it contains all vertices in U that neighbor at least $(1-\epsilon)$ of the vertices in $\hat{W}$).

We refer to such an approximate biclique (cluster) as an ϵ-biclique (ϵ-cluster). We give an algorithm that outputs a subset $\hat{W}$ such that $(\hat{U}, \hat{W})$ has almost as many edges as the optimum biclique (U^*, W^*).

Our algorithm runs efficiently provided that ϵ is constant and the fraction of points in both U^* and W^* is sufficiently large. In other words, that the length of the conjunction and the number of points that satisfy the conjunction is large. Indeed, if $|U^*| \geq \rho_{\mathrm{U}} \cdot |U|$ and $|W^*| \geq \rho_{\mathrm{W}} \cdot |W|$, for certain size parameters $0 < \rho_{\mathrm{U}}, \rho_{\mathrm{W}} \leq 1$, then our algorithm draws a sample of size polynomial in all input parameters, and runs in time linear in $|W|$, *independent* of $|U|$, quasi-polynomial in $1/\rho_{\mathrm{U}}$ and $1/\rho_{\mathrm{W}}$, and exponential in $\log(1/\epsilon)/\epsilon^2$.

While it would be more desirable to have an algorithm with running time polynomial in all problem parameters, we cannot expect to have polynomial dependence in $1/\epsilon$ since in such a case we could use the algorithm to solve the original NP-hard problem in polynomial time by setting $\epsilon < \frac{1}{|U||W|}$. We leave open the question of whether it is possible to obtain an algorithm with polynomial dependence on $1/\rho_{\mathrm{U}}, 1/\rho_{\mathrm{W}}$. This paper addresses the situation when both $1/\rho_{\mathrm{U}}$ and $1/\rho_{\mathrm{W}}$ are small (for example, constant); such a situation has practical motivation. For instance, target marketing schemes are often designed to affect large portions (e.g., 20-30%) of the customer population.

Collection of Large Conjunctive Clusters. We next discuss the more general problem of identifying a collection of large conjunctive clusters. We define the k best conjunctive clusters as the largest conjunctive clusters that don't overlap too much. Given size parameters $0 < \rho_{\mathrm{U}}, \rho_{\mathrm{W}} \leq 1$, and given k the number of clusters, we give an algorithm that outputs k subsets $\hat{W} \subseteq W$ for which the following holds: $|\hat{W}| \geq \rho_{\mathrm{W}} \cdot |W|$; $|\hat{U}| \geq \rho_{\mathrm{U}} \cdot |U|$ (where $\hat{U}$ is as defined above given $\hat{W}$); the different ϵ-bicliques $(\hat{U}, \hat{W})$ don't overlap too much in terms of edges. Further, for every true conjunctive cluster (U', W') such that $|U'| \geq \rho_{\mathrm{U}} \cdot |U|$ and $|W'| \geq \rho_{\mathrm{W}} \cdot |W|$, either there is an ϵ-cluster in our collection that approximately covers (U', W') or (U', W') is smaller (not much bigger) than every ϵ-cluster in the collection. The running time of our algorithm is quasi-polynomial in k, $1/\rho_{\mathrm{U}}$ and $1/\rho_{\mathrm{W}}$, exponential in $\log(1/\epsilon)/\epsilon^2$, linear in $|W|$, and independent of $|U|$.

Finding Approximations to ϵ-bicliques. The above algorithms can also be adapted to finding approximations to large ϵ-bicliques. This may be useful when there is no large "perfect" conjunctive cluster, but there are large ϵ-clusters. The modified algorithm will output an $O(\epsilon^{1/3})$-biclique that is almost as large as the largest ϵ-biclique and can be further extended to output a collection of such large approximate bicliques.

1.3 Related Work

As mentioned previously, the maximum-edge biclique problem is NP-hard [28]. Recently, Feige [12] has shown that under the assumption that refuting 3SAT is hard on average, the maximum-edge biclique problem is hard to approximate to within a certain constant. Furthermore, for certain constants $\alpha < \beta$, it is hard to distinguish between the case in which the maximum biclique has size at least

$\beta \cdot |U| \cdot |W|$ (where $G = (U, W, E)$ is the given bipartite graph) and the case in which the maximum biclique has size less than $\alpha \cdot |U| \cdot |W|$.

Related to the problem of finding almost bicliques is finding dense subgraphs (i.e., with maximum average degree). Finding a densest subgraph of a particular size is NP-hard (since CLIQUE is NP-hard). The algorithm in [30] gives an approximation factor of $O(n^{1/3})$ and the algorithm of [4] gives a PTAS for dense graphs. Finding the densest subgraph (without size constraints) can be performed in polynomial time (*cf.* [16, 7]).

Our algorithms are related to Property Testing algorithms on dense graphs [17] (and in particular are inspired by the CLIQUE-testing algorithm in [17]). Such algorithms are designed to decide whether a given (dense) graph has a certain property or whether many edge modifications should be performed so that it obtains the property. Many testing algorithms can be modified so as to obtain approximate solutions to the corresponding search problems, similarly to the approximate solutions studied in this paper. However, none of the known property testing algorithms (and their extensions to approximation algorithms) directly applies to our problem. In particular, the most general family of graph properties studied in [17] does not capture our definition of clustering which allows for overlapping subsets of vertices. Other related work on approximation algorithms and testing algorithm on dense graphs includes [4, 14, 3].

The general notion of finding cluster descriptions is known as conceptual clustering [24]. Pitt and Reinke [29] show that the Hierarchical Agglomerative Clustering (HAC) algorithm finds an optimum clustering under particular conditions on intra and inter cluster distance. A separate conjunctive clustering problem, considered in [25], was that of finding $k \geq 2$ disjoint conjunctive descriptions $c_1 \ldots, c_k$ such that $\sum_{i=1}^{k} |c_i||Y_i|$ is maximized, and no point satisfies both c_i and c_j. These two results are not applicable to our problem since in particular we do not require that each point be assigned to a cluster, that the clusters be disjoint, or that a point exactly satisfy a conjunction in order to be assigned to it. Another paper on identifying descriptions of clusters, by Agrawal et al [1], gives algorithms for identifying DNF descriptions for each cluster. In this work the objective function is different in that a cluster is a union of connected, high density regions, where a region has high density if it has more density than the area around it.

Research on discovering web communities [23, 15, 13] is also related to CC. A web community is a set of web pages that are all relevant to each other. One way to view the community discovery problem is as a bipartite graph $G = (U, W, E)$ where $U = W$ are the pages on the web and E consists of edges (u, w) if there is a hyperlink from u to w or if $u = w$. A biclique (U', W') forms a community since each page in U' is linked to each page in W'. Our results can be used to identify a good cover of the *large* communities on the web. In contrast, our algorithms are not designed to find small communities, also known as "cores" as studied by [23], where the goal is to for example find all $K_{3,2}$'s.

The frequent itemset problem [2, 19] is also closely related to CC. Given a collection of points P in $\{0, 1\}^d$, the frequent itemset problem is that of identi-

fying all subsets of variables that have high support, i.e., all subsets of variables that satisfy a sufficiently large fraction of P. A large conjunctive cluster is in some sense a maximally frequent itemset. The key difference between the two is that whereas in the frequent set formulation the identification of a border separating the frequent from the infrequent sets is critical, our objective is to find a collection of k conjunctions that don't overlap too much and that "dominate" all the big conjunctions.

2 Preliminaries and Problem Definitions

As noted in the introduction, it will be convenient to define our problems using a graph-theoretic formulation. Given a bipartite graph $G = (U, W, E)$ and two subsets $U' \subseteq U$ and $W' \subseteq W$, we denote by $E(U', W')$ the subset of all edges between vertices in U' and vertices in W'. That is, $E(U', W') \stackrel{\text{def}}{=} \{(u, w) \in E : u \in U', w \in W'\}$. We refer to such a pair (U', W') as a *bisubgraph*. For a vertex v we denote the neighbor set of v by $\Gamma(v)$. For a subset S of vertices, we let $\Gamma(S) \stackrel{\text{def}}{=} \cap_{v \in S} \Gamma(v)$ denote the set of vertices that neighbor *every* vertex in S. For a subset S and a parameter $\epsilon < 1$, we let $\Gamma_\epsilon(S) \stackrel{\text{def}}{=} \{w : |\Gamma(w) \cap S| \geq (1-\epsilon)|S|\}$ denote the set of vertices that neighbor all but an ϵ-fraction of S.

Definition 1. *Given a bipartite graph $G = (U, W, E)$, a bisubgraph (U', W') is a* **biclique** *if $E(U', W') = U' \times W'$. That is, $W' \subseteq \Gamma(U')$. The (edge-)***size** *of a biclique (U', W') is $|E(U', W')| = |U'| \cdot |W'|$, and a* **maximum biclique** *is a biclique (U', W') for which $|U'| \cdot |W'|$ is maximized over all bicliques.*

Note that if we have one side of the biclique W^* then we can obtain the other side of the biclique U^* since $U^* = \Gamma(W^*)$ (and vice versa). This implies that outputting the conjunctive description W^* suffices for identifying the cluster.

As noted previously, the maximum biclique problem is NP-hard. Here we suggest a relaxation of the maximum biclique problem which allows the output to be close to a biclique.

Definition 2. *We say that (U', W') is* **ϵ-close** *to being a biclique, for $0 \leq \epsilon \leq 1$, if every vertex in U' neighbors at least $(1-\epsilon)$ of the vertices in W'. For the sake of succinctness, we say that (U', W') is an* **ϵ-biclique**. *The size of an ϵ-biclique is $|E(U', W')|$ (which is $\geq |U'| \cdot |W'| \cdot (1-\epsilon)$).*

In the context of conjunctive clusters, an ϵ-biclique corresponds to a pair (Y, c) such that every point in Y satisfies most (at least $(1-\epsilon)$) of the attributes in c. Note that the asymmetry between U' and W' in the definition of an ϵ-biclique corresponds to our needs in the context of clustering where the two sides of the ϵ-biclique in fact have a different role. Similarly to the biclique case, if we discover W' then U' is completely determined, i.e., $(U', W') = (\Gamma_\epsilon(W'), W')$. This is especially useful in the context of clustering since $W' = c$ is the description of the cluster, and so we do not need to output explicitly all points Y in the cluster.

Our first problem formulation follows.

Problem 1. Given a bipartite graph $G = (U, W, E)$, find a subset $W' \subseteq W$ such that the ϵ-biclique $(\Gamma_\epsilon(W'), W')$ is at least $(1-b\epsilon)$ times as large as the maximum biclique for a small constant b.

For our sublinear result, $b = 2$. If we are allowed time linear in $|U|$, then we can show that $b = 0$.

Collections of Large Bicliques. The above relaxation addresses the issue of finding a *single* approximate maximum biclique. We now turn to defining a good *collection* of at most k bicliques where k is a given parameter.

As in the case of a single biclique, we would like the bicliques in the collection to be large. On the other hand, the number of bicliques in the collection should be bounded. Therefore, if there are several large bicliques that are very similar, we may prefer including only one of them in the collection as a "representative", so as to allow ourselves to include other bicliques that are possibly smaller but less similar. We next introduce the notion of coverage.

Definition 3. *Let $G = (U, W, E)$ be a bipartite graph and let $U', U'' \subseteq U$ and $W', W'' \subseteq W$. We say that (U', W') is a* **δ-cover** *of (U'', W'') if*

$$\frac{|E(U'', W'') \setminus E(U', W')|}{|E(U'', W'') \cup E(U', W')|} \leq \delta \, .$$

We next define *domination* which essentially states that a subgraph (U', W') is dominated by a collection of subgraphs C if there is a subgraph in C that covers (U', W') or if every pair in C is either larger or only slightly smaller than (U', W').

Definition 4. *Let $G = (U, W, E)$ be a bipartite graph and let $\mathcal{C} = \{(U_i, W_i)\}_{i=1}^{k}$ be a collection of pairs of vertex subsets where $U_i \subseteq U$, and $W_i \subseteq W$. We say that $\mathcal{C}$* **(δ,ϵ)-dominates** *a pair (U', W') if either there exists a pair $(U_i, W_i) \in \mathcal{C}$ that δ-covers (U', W'), or $|E(U', W')| \leq (1+\epsilon) \cdot \min_j\{|E(U'_j, W'_j)|\}$.*

We will also sometimes say that a collection C_1 of subgraphs dominates another collection of subgraphs C_2 if C_1 dominates each subgraph $(U_2, W_2) \in C_2$.

The following definition ensures that the collection of subgraphs output by the algorithm don't overlap with each other too much.

Definition 5. *Let $G = (U, W, E)$ be a bipartite graph and let $\mathcal{C} = \{(U_i, W_i)\}_{i=1}^{k}$ be a collection of pairs of vertex subsets where $U_i \subseteq U$, and $W_i \subseteq W$. We say that $\mathcal{C}$ is* **δ-diverse** *if for every two different pairs (U_i, W_i) and (U_j, W_j) in $\mathcal{C}$, neither is a δ-cover of the other.*

Since this paper is focused on identifying large conjunctive clusters, we introduce two lower-bound parameters, ρ_{U} and ρ_{W}, which the algorithm is provided with, and consider only bicliques (U', W') such that $|U'| \geq \rho_{\text{U}} \cdot |U|$ and $|W'| \geq \rho_{\text{W}} \cdot |W|$. These parameters prevent the algorithm from outputting clusters with few points (ρ_{U}) or with little in common (ρ_{W}). Let $\mathcal{B}(\rho_{\text{U}}, \rho_{\text{W}})$ denote the set of all bicliques (U', W') in G such that $|U'| \geq \rho_{\text{U}} \cdot |U|$ and $|W'| \geq \rho_{\text{W}} \cdot |W|$.

Given the above definitions, a natural problem is to find a collection of at most k bicliques in $\mathcal{B}(\rho_{\text{U}}, \rho_{\text{W}})$, that is both δ-diverse and (δ, ϵ)-dominates every $(U', W') \in \mathcal{B}(\rho_{\text{U}}, \rho_{\text{W}})$. Here we define a relaxation:

Problem 2. Let $G = (U, W, E)$ be a given bipartite graph, $0 < \rho_{\text{U}}, \rho_{\text{W}} \leq 1$ two size parameters, k an integer, $0 \leq \delta \leq 1$ a diversity/covering parameter, and $0 \leq \epsilon \leq 1$ an approximation parameter. Find a collection $\tilde{\mathcal{C}}$ of at most k ϵ-bicliques in $\mathcal{B}_\epsilon(\rho_{\text{U}}, \rho_{\text{W}})$ such that $\tilde{\mathcal{C}}$ is δ-diverse and for every $(U', W') \in \mathcal{B}(\rho_{\text{U}}, \rho_{\text{W}})$, (U', W') is $(b \cdot (\delta + \epsilon), b' \cdot \epsilon)$-dominated by $\tilde{\mathcal{C}}$ for some small constants b and b'.

For our sublinear result, $b = 4$ and $b' = 2$. If we are given time linear in $|U|$ then we can show that $b = 3$ and $b' = 0$.

Note that Problem 2 allows clusters to overlap in the sense that a given point may belong to multiple clusters. Note also that the problem definition doesn't require that all points be clustered. These two facts are an interesting contrast to clustering problem formulations that require a strict partition of the points [21, 8, 11, 18, 22, 6].

3 A Good Seed

In this section we discuss a central building block of our algorithms. Consider a fixed biclique (U^*, W^*) and assume it is maximal. As noted in the preliminaries section, if we knew U^* we could obtain W^* exactly by simply considering $\Gamma(U^*)$. We can then get U^* back by considering $\Gamma(\Gamma(U^*))$. Clearly we do not have U^*, or else we would be done. Suppose instead, as a mental experiment, that we were able to obtain a (small) *random sample* S from U^*. Then for every sample $S \subset U$, W^* is contained in $\Gamma(S)$. However, $\Gamma(S)$ may contain many additional vertices outside of W^*. As a consequence, if we now take $\Gamma(\Gamma(S))$ we may get a very small subset (or even an empty set). However, as we show below, if we instead take $\Gamma_\epsilon(\Gamma(S))$, then with high probability over the choice of a sufficiently large sample S, the ϵ-biclique $(\Gamma_\epsilon(\Gamma(S)), \Gamma(S))$ is at least as large as the biclique (U^*, W^*).

We think of the sample S as being a "good seed" for the biclique (U^*, W^*). In the next section we shall get rid of the imaginary assumption that we can directly sample from U^* in order to obtain the good seed. Let ρ_{W} be a lower bound on $|W^*|/|W|$, and let $\hat{m} = \frac{16}{\epsilon^2} \log \frac{40}{\rho_{\text{W}} \epsilon}$.

Good Seed Algorithm
1. $S \leftarrow$ sample from U^* of size $\hat{m}$
2. $\hat{W} \leftarrow \Gamma(S)$
3. Output $\hat{W}$

Lemma 1. *Let $\hat{W}$ be as constructed in the Good Seed Algorithm on a sample S of size $\hat{m}$ drawn uniformly from U^*. With probability at least $\frac{9}{10}$ over the choice of $S \subset U^*$, $|E(\Gamma_\epsilon(\hat{W}), \hat{W})| \geq |U^*| \cdot |W^*|$.*

In order to prove the lemma, it will be helpful to partition the vertices in $\hat{W}$, i.e., the intersection of the neighbors of S, into those that are in the optimum biclique, W^*, those that neighbor a significant fraction of U^* (H, for high degree) and those that don't (L, for low degree). Note again that since S is a subset of U^*, $\hat{W}$ must contain all of W^*. We'll show that with high probability there are very few L vertices. Thus, since most of the vertices in $\hat{W}$ are either in the optimum biclique or have high degree with U^*, we'll argue that the bisubgraph $(\Gamma_\epsilon(\hat{W}), \hat{W}))$ has at least as many edges as the optimum. We now precisely define the terms High and Low.

Definition 6. *We say that a vertex $w \in W$ has* **high degree with respect to** U^* *if*

$$\frac{|\Gamma(w) \cap U^*|}{|U^*|} \geq 1 - (\epsilon/4)^2 \, .$$

Otherwise it has **low degree with respect to** U^**.*

Note that in particular, every $w \in W^*$ has high degree with respect to U^* (since for every $w \in W^*$, $\frac{|\Gamma(w) \cap U^*|}{|U^*|} = 1$). We will be interested in samples of U^* that are "good seeds": we would like a sample S so that $\Gamma(S)$ contains W^* plus mostly vertices that have high degree with respect to U^*.

Definition 7. *We say that a subset $S \subseteq U^*$ is* **a good seed of** U^* *if the number of vertices in $\Gamma(S) \subseteq W$ that have low degree with respect to U^* is at most $(\epsilon/4)|W^*|$.*

We now claim that our sample S is sufficiently large to ensure that $\Gamma(S)$ has few low degree vertices with respect to U^*, i.e., that S is a good seed of U^*. The lemma (proof in [26]) can be proved via probabilistic techniques.

Lemma 2. *With probability at least $\frac{9}{10}$ the sample S drawn in step 1 of the Good Seed Algorithm is a good seed of U^*.*

We next show that if S is a good seed of U^* then the ϵ-biclique $(\Gamma_\epsilon(\hat{W}), \hat{W}) = (\Gamma_\epsilon(\Gamma(S)), \Gamma(S))$ has as many edges as (U^*, W^*).

Lemma 3. *Let $\hat{W}$ be as constructed in the Good Seed Algorithm on a sample S of size $\hat{m}$ drawn from U^*. If S is a good seed of U^* then $(\Gamma_\epsilon(\hat{W}), \hat{W})$ is an $\epsilon/4$-cover of (U^*, W^*) and $|E(\Gamma_\epsilon(\hat{W}), \hat{W})| \geq |U^*| \cdot |W^*|$.*

Proof. The subset $\hat{W}$ consists of three parts: (1) the vertices of W^*; (2) a subset of vertices, denoted H, that have high degree with respect to U^*; (3) a subset of vertices, denoted L, having low degree with respect to U^*. We will show that most of U^* neighbors $\hat{W}$ by considering two cases based on whether $|H|$ is small or large. In what follows, let $\hat{U} \stackrel{\text{def}}{=} \Gamma_\epsilon(\hat{W})$.

$|H| \leq \frac{\epsilon}{2}|W^*|$: If W^* accounts for at least $(1-\epsilon)$ of $\hat{W}$ then by Step 3 of the algorithm, U^* will be part of $\hat{U}$. Indeed this is true:

$$\frac{|W^*|}{|\hat{W}|} = \frac{|W^*|}{|H| + |L| + |W^*|} > \frac{|W^*|}{\frac{\epsilon}{2}|W^*| + \frac{\epsilon}{4}|W^*| + |W^*|} = \frac{1}{1 + \frac{\epsilon}{2} + \frac{\epsilon}{4}} \geq (1-\epsilon) \quad (1)$$

(recall that $|L| < \frac{\epsilon}{4}|W^*|$ since S is a good seed). Since $U^* \subseteq \hat{U}$ and $W^* \subseteq \hat{W}$ we have that $(\hat{U}, \hat{W})$ completely covers (U^*, W^*) and so $|E(\hat{U}, \hat{W})| \geq |U^*||W^*|$ as desired.

$|H| > \frac{\epsilon}{2}|W^*|$: We first show that all but at most an $\epsilon/4$-fraction of the vertices in U^* have at least $(1-\epsilon/4)|H|$ neighbors in H. Let the subset of vertices in U^* having at least $(1-\epsilon/4)|H|$ neighbors in H be denoted Q^*. Thus we would like to show that $|Q^*| \geq (1-\epsilon/4)|U^*|$.

Let α be such that $|Q^*| = (1-\alpha)|U^*|$. Assume contrary to the claim, that $\alpha > \epsilon/4$. Then the total number of edges between U^* and H would be less than:

$$\begin{aligned} &|Q^*| \cdot |H| + |U^* \setminus Q^*| \cdot (1-\epsilon/4)|H| \\ &= (1-\alpha)|U^*| \cdot |H| \ + \ \alpha|U^*| \cdot (1-\epsilon/4)|H| \\ &= (1-\alpha+\alpha-\alpha\cdot(\epsilon/4))|U^*| \cdot |H| \ < \ (1-(\epsilon/4)^2)|U^*| \cdot |H| \ . \end{aligned} \tag{2}$$

But by definition of H, $|E(U^*, H)| \geq |H| \cdot (1-(\epsilon/4)^2)|U^*|$, and we have reached a contradiction.

Since $|L| \leq (\epsilon/4)|W^*|$, every vertex in Q^* has at least $(1-\epsilon)|\hat{W}|$ neighbors in $\hat{W}$, and hence $Q^* \subseteq \hat{U}$. Since we have shown that $|Q^*| \geq (1-\epsilon/4)|U^*|$ (where $Q^* \subseteq U^*$), we have that $E(\hat{U}, \hat{W})$ contains all edges in $E(U^*, W^*)$ but at most $(\epsilon/4) \cdot |U^*| \cdot |W^*|$ And so $(\hat{U}, \hat{W})$ certainly $(\epsilon/4)$-covers (U^*, W^*). Finally, by definition of Q^* and what we have shown concerning its size,

$$\begin{aligned} |E(Q^*, \hat{W})| &\geq |Q^*| \cdot (|W^*| + (1-\epsilon/4)|H|) \\ &\geq (1-\epsilon/4)|U^*| \cdot (|W^*| + |H|(1-\epsilon/4)) \\ &> |U^*| \cdot |W^*| \cdot (1-\epsilon/4) \cdot (1+\epsilon/2(1-\epsilon/4)) \ > \ |U^*| \cdot |W^*| \end{aligned} \tag{3}$$

Since $|E(\hat{U}, \hat{W})| \geq |E(Q^*, \hat{W})|$, we are done. □

The proof of Lemma 1 directly follows from Lemmas 2 and 3.

4 Conjunctive Clustering Algorithm

We now turn to the problem of identifying conjunctive clusters. We begin by considering the problem of finding one large approximate conjunctive cluster. Then we consider finding a good collection of them.

4.1 Approximate Maximum Biclique

Given ρ_{U} and ρ_{W} (for which $\mathcal{B}(\rho_{\mathrm{U}}, \rho_{\mathrm{W}})$ is non-empty) we shall show how to find an ϵ-biclique in which the number of edges is almost as large as in a maximum biclique in $\mathcal{B}(\rho_{\mathrm{U}}, \rho_{\mathrm{W}})$.[4] Solving this problem is interesting in its own right and the solution is also later used to identify k conjunctive clusters.

[4] If the algorithm is not provided with lower bounds ρ_{U} and ρ_{W} then it can search for them using a standard doubling process.

Since we cannot actually sample from the left-hand-side U^*, we instead use what is sometimes referred to as *exhaustive sampling* (see e.g. [9, 4, 17, 14]). Namely, we sample from U, and consider all subsets of the sample whose size is lower bounded by a certain threshold. It can then be verified that if the sample is sufficiently large, then with high probability one of these subsets is a good seed. However, now we have to address a new problem: How do we decide which subset is the good seed? We could of course check the resulting ϵ-biclique for each subset, but this would take time linear in U, and we are interested in an algorithm having time *independent* of $|U|$. As one may guess at this point, we solve this by sampling again from U.

Let $\hat{m}$ be as defined in the Good Seed Algorithm. Also, let $m = m(\epsilon, \rho_{\mathrm{U}}, \rho_{\mathrm{W}}) = \frac{2}{\rho_{\mathrm{U}}} \cdot \hat{m}$, and let $t = t(\epsilon, \rho_{\mathrm{U}}) = \frac{96}{\rho_{\mathrm{U}} \cdot \epsilon^2} \cdot m$.

Algorithm Approximate Maximum Biclique

1. Draw a sample X of m vertices uniformly and independently from U.
2. Draw another sample T of t vertices uniformly and independently from U.
3. For each subset S of X that has size $\hat{m}$ do:
   (a) $\hat{W}(S) \leftarrow \Gamma(S)$
   (b) $\hat{T}(S) \leftarrow$ vertices in T that neighbor most of $\hat{W}$, i.e., $T \cap \Gamma_\epsilon(\hat{W}(S))$.
4. Among all subsets S considered by the algorithm for which $\hat{T}(S) \geq (3\rho_{\mathrm{U}}/4)t$, let Z be the one for which $|\hat{T}(Z)| \cdot |\hat{W}(Z)|$ is maximized. Output $\hat{W}(Z)$.

Let (U^*, W^*) be a maximum biclique. For any subset S, let $\hat{W}(S) = \Gamma(S)$ and let $\hat{U}(S) = \Gamma_\epsilon(\hat{W}(S))$. Let $\hat{G}(S) = (\hat{U}(S), \hat{W}(S))$ be the bisubgraph determined by S. We define the *true relative size* of $\hat{G}(S)$ to be $(|\hat{U}(S)| \cdot |\hat{W}(S)|)/(|U| \cdot |W|)$ and the *estimated relative size* of $\hat{G}(S)$ to be $(|\hat{T}(S)| \cdot |\hat{W}(S)|)(t \cdot |W|)$. (Recall that $|T| = t$.) We also define a *good subset* S_g to be one for which $|\hat{U}(S_g)|/|U| \geq \rho_{\mathrm{U}}/2$ and a *bad subset* S_b to be one for which $|\hat{U}(S_b)|/|U| < \rho_{\mathrm{U}}/2$.

The algorithm works via the following reasoning. We show that one of the subsets S considered in step 3 of the algorithm is a good seed of U^* with high probability. We denote this subset by S^*. By Lemma 3, we will then know that the bisubgraph $(\Gamma_\epsilon(\hat{W}(S^*)), \hat{W}(S^*))$ has at least as many edges as the optimum. We then show that, with high probability, the algorithm won't consider any bad subset S_b (since for bad subsets, $\hat{T}(S_b)$ will be too small). On the other hand, the estimated relative size of $\hat{G}(S_g)$ for any good subset S_g is close to its true relative size. Thus in particular, the estimated relative size of $\hat{G}(S^*)$ for the seed S^* (which is a good subset), is close to its true relative size. It will then follow that for $\hat{W}(Z)$ output by the algorithm, the bisubgraph $(\hat{U}(Z), \hat{W}(Z))$ must have about as many (not much fewer) edges as the optimum true maximum biclique (U^*, W^*).

Theorem 1. *With probability at least* $2/3$*, Algorithm Approximate Maximum Biclique outputs a subset* $\hat{W} = \hat{W}(Z)$ *so that* $|E(\Gamma_\epsilon(\hat{W}), \hat{W}))| \geq (1 - 2\epsilon) \cdot |U^*| \cdot |W^*|$*. The running time of the algorithm is exponential in* $\frac{\log(1/\epsilon)}{\epsilon^2}$*, quasi-polynomial in* $\frac{1}{\rho_U}$ *and* $\frac{1}{\rho_W}$*, linear in* $|W|$ *and independent of* $|U|$*.*

Proof. It can be shown via multiplicative Chernoff bounds, that the following holds with probability at least 9/10: (1) One of the subsets considered in step 3 of Algorithm Approximate Maximum Biclique is a good seed. (2) In step 4 of Algorithm Approximate Maximum Biclique, no bad subset X of size $\hat{m}$ will be considered. (3) In step 4 of Algorithm Approximate Maximum Biclique, for any good subset S_g of X of size $\hat{m}$,

$$(1-\epsilon/4)\frac{|\hat{U}(S_g)|}{|U|} \leq \frac{|\hat{T}(S_g)|}{t} \leq (1+\epsilon/4)\frac{|\hat{U}(S_g)|}{|U|}$$

For the rest of the proof, assume that these events in fact happen.

Now we show that the specific subset S^* will not (with high probability) be excluded by Step 4 of the algorithm. First observe that the subset S^* is good since $|\hat{U}(S^*)| \geq (1-\epsilon/4)|U^*| > (\rho_{\mathrm{U}}/2)|U|$. Further, it can be shown that $\hat{T}(S^*)/t \geq 3\rho_{\mathrm{U}}/4$ since $\epsilon < 1/2$:

$$|\hat{T}(S^*)|/t \geq (1-\epsilon/4)|\hat{U}(S^*)|/|U| \geq (1-\epsilon/4)(1-\epsilon/4)\rho_{\mathrm{U}} \geq (1-\epsilon/2)\rho_{\mathrm{U}} \geq 3\rho_{\mathrm{U}}/4$$

and thus S^* will be considered by the algorithm. By the proof of Lemma 3, we have that

$$\frac{|\hat{T}(S^*)| \cdot |\hat{W}(S^*)|}{t \cdot |W|} \geq (1-\epsilon/4) \cdot \frac{|\hat{U}(S^*)| \cdot |\hat{W}(S^*)|}{|U| \cdot |W|} \,. \tag{4}$$

Next we show that the number of edges in the bisubgraph output by the algorithm $(\hat{U}(Z), \hat{W}(Z))$ is not much smaller than the number of edges in the maximum biclique (U^*, W^*).

$$\begin{aligned}
|E(\hat{U}(Z), \hat{W}(Z))| &\geq (1-\epsilon)|\hat{U}(Z)| \cdot |\hat{W}(Z)| && (5)\\
&\geq \frac{1-\epsilon}{1+\epsilon/4} \cdot \frac{|U|}{t} \cdot |\hat{T}(Z)| \cdot |\hat{W}(Z)| && (6)\\
&\geq \frac{1-\epsilon}{1+\epsilon/4} \cdot \frac{|U|}{t} \cdot |\hat{T}(S^*)| \cdot |\hat{W}(S^*)| && (7)\\
&\geq \frac{(1-\epsilon)(1-\epsilon/4)}{1+\epsilon/4} \cdot |\hat{U}(S^*)| \cdot |\hat{W}(S^*)| && (8)\\
&\geq (1-2\epsilon)|U^*| \cdot |W^*| && (9)
\end{aligned}$$

The bound on the running time follows from the fact that we enumerate over all subsets of size $\hat{m}$ of the m vertices drawn in Step 1. The total number of such subsets is $\binom{m}{\hat{m}}$. For each subset S we compute $\Gamma(S)$ and $T \cap \Gamma_\epsilon(\Gamma(S))$. Thus for each subset S, the algorithm spends time $O(t \cdot |W|)$. Hence the total running time is $O(m^{\hat{m}}|W|t|) = O((\frac{1}{\rho_{\mathrm{U}}\epsilon^2}\log\frac{1}{\rho_{\mathrm{W}}\epsilon})^{O(\frac{1}{\epsilon^2}\log\frac{1}{\rho_{\mathrm{W}}\epsilon})}|W|(\frac{1}{(\rho_{\mathrm{U}}\epsilon^2)^2}\log\frac{1}{\rho_{\mathrm{W}}\epsilon}))$. □

4.2 Conjunctive Clustering

Recall that given $\rho_{\mathrm{U}}, \rho_{\mathrm{W}}$, k, ϵ and δ, our goal is to output a collection $\tilde{\mathcal{C}}$ of k ϵ-bicliques that is δ-diverse and that $(b(\delta+\epsilon), b'\epsilon)$-dominates every biclique in $\mathcal{B}(\rho_{\mathrm{U}}, \rho_{\mathrm{W}})$ for small constants b and b'.

We reset $\hat{m}, m$ and t as follows: $\hat{m} = \hat{m}(k, \epsilon, \rho_{\mathrm{U}}, \rho_{\mathrm{W}}) = \Theta\left(\frac{1}{\epsilon^2} \log \frac{k}{\rho_{\mathrm{W}} \cdot \epsilon}\right)$, $m = m(k, \epsilon, \rho_{\mathrm{U}}, \rho_{\mathrm{W}}) = \frac{2 \log k}{\rho_{\mathrm{U}}} \cdot \hat{m}$, and $t = t(k, \epsilon, \rho_{\mathrm{U}}, \rho_{\mathrm{W}}) = \Theta\left(\frac{\log(1/\epsilon)}{\rho_{\mathrm{U}} \cdot \rho_{\mathrm{W}} \cdot \epsilon^3}\right) \cdot m$.

Conjunctive Clustering Algorithm

1. Draw a sample X of m vertices uniformly and independently from U.
2. Draw another sample T of t vertices uniformly and independently from U. Let $\hat{\mathcal{W}} \leftarrow \emptyset$.
3. For each subset S of X that has size $\hat{m}$ do
   (a) $\hat{W}(S) \leftarrow \Gamma(S)$.
   (b) $\hat{T}(S) \leftarrow T \cap \Gamma_\epsilon(\hat{W}(S))$.
   (c) If $|\hat{W}(S)| \geq \rho_{\mathrm{W}} \cdot |W|$ and $|\hat{T}(S)| \geq (\rho_{\mathrm{U}}/2) \cdot t$ then add $\hat{W}(S)$ to $\hat{\mathcal{W}}$.
4. Order the subsets $\hat{W}(S)$ in $\hat{\mathcal{W}}$ according to the magnitude of $|\hat{T}(S)| \cdot |\hat{W}(S)|$. Perform the following at most k times: Add to $\tilde{\mathcal{W}}$ the next subset $\hat{W}(S)$ (according to the above order) such that $(\hat{T}(S), \hat{W}(S))$ is not yet $(\delta + 2\epsilon)$-covered by any $(\hat{T}(S'), \hat{W}(S'))$ where $\hat{W}(S') \in \hat{\mathcal{W}}$.

The next theorem establishes that our algorithm works as desired. Due to space constraints its proof appears in the full version of this paper [26].

Theorem 2. *With probability at least* 4/5 *Algorithm Conjunctive Clustering outputs a collection* $\tilde{\mathcal{W}}$ *of at most* k ϵ*-bicliques such that* $\tilde{\mathcal{C}} = \left\{\left(\Gamma_\epsilon(\hat{W}), \hat{W}\right) : \hat{W} \in \tilde{\mathcal{W}}\right\}$ *is* δ*-diverse, and* $\tilde{\mathcal{C}}$ $((2\delta + 4\epsilon), 2\epsilon)$*-dominates every biclique in* $\mathcal{B}(\rho_U, \rho_W)$*. The running time of the algorithm is exponential in* $\frac{\log 1/\epsilon}{\epsilon^2}$*, quasi-polynomial in* k*,* $1/\rho_U$*, and* $1/\rho_W$ *linear in* $|W|$ *and independent of* $|U|$*.*

An algorithm that runs in time independent of both $|U|$ and $|W|$ can be found in the full version of this paper [26].

5 Finding Approximate ϵ-Bicliques

In Section 4.1 we showed that if the graph contains a large biclique (U^*, W^*), then we can find a subset $\hat{W}$ such that $|E(\Gamma_\epsilon(\hat{W}), \hat{W})| \geq (1 - 2\epsilon)|U^*| \cdot |W^*|$. Let us define a *strong* ϵ-biclique to be a pair (U', W') such that every vertex in U' neighbors at least $(1 - \epsilon)$ of the vertices in W' and every vertex in W' neighbors at least $1 - \epsilon$ of the vertices in U'. Thus this is a strengthening of the definition of an ϵ-biclique. Suppose we know that there exists a strong ϵ-biclique $(\tilde{U}, \tilde{W})$ such that $|\tilde{U}| \geq \rho_{\mathrm{U}} \cdot |U|$ and $|\tilde{W}| \geq \rho_{\mathrm{W}} \cdot |U|$ (but there isn't necessarily such a large biclique). We next show how the Approximate Maximum Biclique algorithm can be modified so as to obtain an $O(\epsilon^{1/2})$-biclique $(\hat{U}, \hat{W})$ such that $|E(\hat{U}, \hat{W})| \geq (1 - O(\epsilon^{1/2}))|E(\tilde{U}, \tilde{W})|$. By small modifications it is possible to deal with the case in which $(\tilde{U}, \tilde{W})$ is an ϵ-biclique (i.e., not a strong one). The extension of finding a collection of large $O(\epsilon^{1/2})$-bicliques is done analogously to what is described in Section 4.2.

Let $\hat{m}$ and m be as in the Approximate Maximum Biclique algorithm, where we assume that $\rho_{\mathrm{U}} \cdot |U|$ and $\rho_{\mathrm{W}} \cdot |W|$ are lower bounds on the sizes of $\tilde{U}$ and $\tilde{W}$, respectively. We also assume that ϵ is sufficiently small ($\epsilon < 1/12$), or else we can replace each occurrence of ϵ with $\min\{\epsilon, 1/12\}$.

Algorithm Approximate Maximum ϵ-Biclique

1. Draw a sample X of m vertices uniformly and independently from U.
2. Draw another sample T of t vertices uniformly and independently from U.
3. For each subset S of X that has size $\hat{m}$ do:
   (a) $\hat{W}(S) \leftarrow \Gamma_{2\epsilon}(S)$
   (b) $\hat{T}(S) \leftarrow T \cap \Gamma_{2\sqrt{\epsilon}}(\hat{W}(S))$.
4. Among all subsets S considered by the algorithm for which $\hat{T}(S) \geq (3\rho_{\mathrm{U}}/4)t$, let Z be such that $|\hat{T}(Z)| \cdot |\hat{W}(Z)|$ is maximized. Output $\hat{W}(Z)$.

The proof of the following theorem can be found in the full version [26].

Theorem 3. *With probability at least* $2/3$*, Algorithm Approximate Maximum* ϵ*-biclique outputs a subset* $\hat{W} = \hat{W}(S)$ *such that* $|E(\Gamma_{2\sqrt{\epsilon}}(\hat{W}), \hat{W})| \geq (1 - 3\sqrt{\epsilon}) \cdot \tilde{\rho}|U||W|$*, where* $\tilde{\rho}|U||W|$ *is the size of a maximum strong* ϵ*-biclique.*

6 Future Work

Some interesting problems left open by our research include: (1) Algorithms with polynomial dependence on $\frac{1}{\rho_{\mathrm{U}}}$ and $\frac{1}{\rho_{\mathrm{W}}}$; (2) Variations where the vertices or edges are weighted; (3) Alternate definitions of k best conjunctive clusters; and (4) Extensions to descriptions that are not conjunctive in nature, e.g., disjunctions.

References

1. R. Agrawal, J.E. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of SIGMOD*, pages 94–105, 1998.
2. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of SIGMOD*, pages 207–216, 1993.
3. N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. *Combinatorica*, 20:451–476, 2000.
4. S. Arora, D. Karger, and M Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *Journal of Computer and System Sciences*, 58:193–210, 1999.
5. Arya, Garg, Khandekar, Munagala, and Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of STOC*, 2001.
6. N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In *Proceedings of FOCS*, pages 938–247, 2002.
7. M. Charikar. Greedy approximation algorithms for finding dense components in a graph. *Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 84–95, 2000.
8. M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of FOCS*, pages 378–388, 1999.
9. W. Fernandez de la Vega. MAX-CUT has a randomized approximation scheme in dense graphs. *Random Structures and Algorithms*, 8:187–198, 1996.

10. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society series B*, 39:1–38, 1977.
11. T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *Proceedings of STOC*, pages 434–444, 1988.
12. U. Feige. Average case complexity and approximation complexity. In *Proceedings of STOC*, 2002.
13. G. Flake, S. Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Proceedings of KDD*, pages 150–160, 2000.
14. A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.
15. D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext*, Structural Queries, pages 225–234, 1998.
16. A. V. Goldberg. Finding a maximum density subgraph. *UC Berkeley Tech Report, CSD-84-171*, 1984.
17. O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.
18. T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2-3):293–306, June 1985.
19. D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen. Data mining, hypergraph transversals, and machine learning (extended abstract). In *Proceedings of PODS*, pages 209–216, 1997.
20. D. Hochbaum and D. Shmoys. A unified approach to approximate algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, July 1986.
21. N. Jain and V.V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proceedings of FOCS*, pages 2–13, 1999.
22. R. Kannan, S. Vempala, and A. Vetta. On clusterings — good, bad and spectral. In IEEE, editor, *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 367–377, 2000.
23. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1481–1493, May 1999.
24. R. S. Michalski. Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. Technical Report 1026, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1980.
25. N. Mishra, D. Oblinger, and L. Pitt. Sublinear time approximate clustering. In *Proceedings of SODA*, pages 439–447, 2001.
26. N. Mishra, D. Ron, and R. Swaminathan. Large conjunctive clusters and bicliques. Available from the authors, 2002.
27. R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric k-clustering. In IEEE, editor, *41st Annual Symposium on Foundations of Computer Science*, pages 349–358, 2000.
28. R. Peeters. The maximum edge biclique problem is NP-complete. Unpublished manuscript, 2000.
29. L. Pitt and R.E. Reinke. Criteria for polynomial-time (conceptual) clustering. *Machine Learning*, 2:371, 1987.
30. D. Peleg U. Feige, G. Kortsarz. The dense-k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.