Efficient Algorithms for Generating Provably Near-Optimal Cluster Descriptors for Explainability

Prathyush Sambaturu,[1] **Aparna Gupta,**[2] **Ian Davidson,**[3]
S. S. Ravi,[4,5] **Anil Vullikanti,**[1,4] **Andrew Warren**[4]
[1]Department of Computer Science, University of Virginia, Charlottesville, VA 22904
[2]Department of Computer Science, Virginia Tech, Blacksburg, VA 24060
[3]Department of Computer Science, UC Davis, Davis, CA 95616
[4]Biocomplexity Institute and Initiative, University of Virginia, Charlottesville, VA 22904
[5]Department of Computer Science, University at Albany–SUNY, Albany, NY 12222
{pks6mk, vsakumar, asw3xp}@virginia.edu, agupta12@vt.edu, davidson@cs.ucdavis.edu, ssravi0@gmail.com

Abstract

Improving the explainability of the results from machine learning methods has become an important research goal. Here, we study the problem of making clusters more interpretable by extending a recent approach of [Davidson et al., NeurIPS 2018] for constructing succinct representations for clusters. Given a set of objects S, a partition π of S (into clusters), and a universe T of tags such that each element in S is associated with a subset of tags, the goal is to find a representative set of tags for each cluster such that those sets are pairwise-disjoint and the total size of all the representatives is minimized. Since this problem is **NP**-hard in general, we develop approximation algorithms with provable performance guarantees for the problem. We also show applications to explain clusters from datasets, including clusters of genomic sequences that represent different threat levels.

1 Introduction

As AI and machine learning (ML) methods become pervasive across all domains from health to urban planning, there is an increasing need to make the results of such methods more interpretable. Providing such explanations has now become a legal requirement in some countries (Goodman and Flaxman 2016). Many researchers are investigating this topic under supervised learning, particularly for methods in deep learning (see e.g., (Proc. XAI-2017 Workshop 2017; Proc. XAI-2018 Workshop 2018)). Clustering is a commonly used unsupervised ML technique (see e.g., (Blondel et al. 2008; Bolla 2013; Fortunato 2010; Tan, Steinbach, and Kumar 2006; Han and Kamber 2011; Zaki and Meira 2014)). It is routinely performed on diverse kinds of datasets, sometimes after constructing network abstractions, and optimizing complex objective functions (e.g., *modularity* (Blondel et al. 2008)). This can often make clusters hard to interpret especially in a post-hoc analysis. Thus, a natural question is whether it is possible to *explain* a given set of clusters, using additional attributes which, crucially, were not used in the clustering procedure. One motivation for our work is to understand the *threat levels* of pathogens for which genomic sequences are available. In (Jain, Gali, and Kihara 2018; Jain and Kihara 2018; Hamid and Friedberg 2018; Ramesh and Warren 2018), researchers have been able to identify some genomic sequences as coming from harmful pathogens (through lab experiments and bioinformatics analysis). Understanding what makes some sequences harmful and distinguishing them from harmless sequences corresponds to the problem of interpreting the clusters.

Davidson et al. (2018) present the following formulation of the *Cluster Description Problem* for explaining a given set of clusters. Let $S = \{s_1, \ldots, s_n\}$ be a set of n objects. Let $\pi = \{C_1, \ldots, C_k\}$ be a partition of S into $k \geq 2$ clusters. Let T be the universe of tags such that each object $s_i \in S$ is associated with a subset $t_i \subseteq T$ of tags. A **descriptor** D_i for a cluster C_i ($1 \leq i \leq k$) is a subset of T. An object s_j in cluster C_i is said to be *covered* by the descriptor D_i if at least one of the tags associated with s_j is in D_i. The goal is to find k pairwise disjoint descriptors (one descriptor per cluster) so that *all* the objects in S are covered and the total number of tags used in all the descriptors, which will henceforth be referred to as the "cost" of the solution, is minimized. Davidson et al. (Davidson, Gourru, and Ravi 2018) showed that even deciding whether there exists a feasible solution (with no restriction on cost) is **NP**-hard. They use Integer Linear Programming (ILP) methods to solve the problem and other relaxed versions (e.g., not requiring coverage of all objects in S, referred to as the "cover-or-forget" version, if there is no *exact* feasible solution) on social media datasets. They point out that this gives interesting and representative descriptions for clusters. However, they leave open the questions of designing efficient and rigorous approximation algorithms, which can scale to much larger datasets, and a deeper exploration of different notions of approximate descriptions for real world datasets.

Our contributions. We find in some datasets that the exact cover formulation of (Davidson, Gourru, and Ravi 2018) does not have a feasible solution. The "cover-or-forget" variation does give a solution, but might have highly unbalanced coverage, i.e., one cluster gets covered well, but not the others. We extend the formulation of (Davidson, Gourru, and Ravi 2018) to address these issues, develop a suite of scal-

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

able algorithms with rigorous guarantees, and evaluate them on several real world datasets. A list of our contributions is given below.

(1) **Formulation.** We introduce the MinConCD problem for cluster description, with *simultaneous* coverage guarantees on all the clusters (defined formally in Section 2). Informally, given a requirement $M_i \leq |C_i|$ for the number of objects to be covered in each cluster C_i, the goal is to find pairwise disjoint descriptors of minimum cost such that at least M_i objects are covered in C_i. MinConCD gives more useful cluster descriptions in several datasets, as we discuss below. However, this problem turns out to be very hard—specifically, we show (in (Sambaturu et al. 2019)) that if the coverage constraints for each cluster must be met, then unless **P = NP**, for any $\rho \geq 1$, there is no polynomial time algorithm that can approximate the cost to within a factor of ρ. Therefore, we consider (α, δ)–approximate solutions, which ensure coverage of at least αM_i objects in cluster C_i $(1 \leq i \leq k)$ using a cost of at most δB^*, where B^* is the minimum cost needed to satisfy the coverage requirements.

(2) **Rigorous algorithms.** We design a randomized algorithm, ROUND, for MinConCD, which is based on rounding a linear programming (LP) solution. We prove that it gives an $(1/8, 2)$–approximation, with high probability. For the special case of $k = 2$, we present an $(1 - 2/e, 1)$-approximation algorithm using techniques from submodular function maximization over a matroid (Section 4).

(3) **Experimental results.** We evaluate our algorithms on real and synthetic datasets (Section 5). We observe that MinConCD gives more appropriate descriptions than those computed using the formulations of (Davidson, Gourru, and Ravi 2018). Our results show that the approximation bounds of ROUND are comparable with the optimum solutions of (Davidson, Gourru, and Ravi 2018) (computed using integer linear programming), and significantly better than the worst-case theoretical guarantees. ROUND also scales well to instances which are over two orders of magnitude larger than those considered in (Davidson, Gourru, and Ravi 2018). The different "knobs" in the formulation (coverage requirement per cluster, cost budget and the allowed overlap) provide a spectrum of descriptions, which allow a practitioner to better explore and understand the clusters. Qualitative analysis of the descriptions gives interesting insights into the clusters. In particular, for the threat sequence dataset, our results give a small set of intuitive attributes which separate the harmful sequences from the harmless ones.

(4) **Extensions and additional algorithmic results.** The theoretical guarantees of ROUND hold only when $M_i = \Theta(|C_i|)$ for each i (i.e., the requirement is to cover a constant fraction of objects in each cluster); the analysis breaks down when $M_i = o(|C_i|)$ for some clusters. When the M_i's are arbitrary, we develop a different randomized rounding algorithm that gives an $(O(1), \eta)$–approximation, where η is the maximum number of objects covered by any tag. Further, we show that ROUND also works for a bounded overlap version of the problem for $k = 2$, where cluster descriptors may overlap. Finally, when $|t_i| \leq \gamma$ for each s_i, we design a simple dynamic programming algorithm that gives an $O(\frac{1}{\gamma}, 1)$-approximation; see (Sambaturu et al. 2019).

Our techniques. While the cluster description problem involves *covering* constraints for each cluster (as in the standard maximum coverage problem (Khuller, Moss, and Naor 1999)), it is much harder because the disjointness requirement leads to independence constraints on the tags to be chosen. A standard approach for approximating covering problems is randomized rounding (see, e.g.,(Srinivasan 1995)). However, here, the events that two objects s_i and $s_{i'}$ are covered are dependent if $t_i \cap t_{i'} \neq \emptyset$; as a consequence, a standard Chernoff bound type concentration analysis cannot be used. We address these issues by observing that the events that s_i and $s_{i'}$ are not covered have a specific type of dependency for which the upper tail bound by Janson and Rucinski (Janson and Rucinski 2002) gives a concentration bound. We develop a new way to analyze our randomized rounding scheme by bounding the number of objects which are not covered in each cluster.

Though coverage is a submodular function (see, e.g., (Calinescu et al. 2011)), it is not clear how to use submodular function maximization techniques to simultaneously maximize the coverage within all clusters. We show that the "saturation" technique of (Krause, McMahan, and Guestrin 2008), which uses the sum of the minimum of each submodular function and a constant, can be adapted for the case $k = 2$, but not for larger k because the problem in (Krause, McMahan, and Guestrin 2008) is for a uniform matroid constraint, whereas here we have partition matroid constraints.

Related work. The topic of "Explainable AI" (Gunning 2017) has attracted a lot of attention especially under supervised learning. In particular, many researchers have studied the topic in conjunction with methods in deep learning (e.g., (Proc. XAI-2018 Workshop 2018)). To our knowledge, not much work has been done in the context of interpreting results from clustering. The topic of allowing a human to interpret a given clustering and provide suggestions for improvement was considered in (Kuo et al. 2017). Their goal was to improve the clustering quality through human guidance, and they used constraint programming techniques to obtain improvements. Other methods for improving a given clustering were considered in (Dang and Bailey 2010; Qi and Davidson 2009). The notion of "descriptive clustering" studied in (Dao et al. 2018) is different from our work; their idea is to allow the clustering algorithm to use both the features of the objects to be clustered and the descriptive information for each object. They present methods for constructing the Pareto frontier based on two objectives, one based on features and the other based on the descriptive information. Like (Davidson, Gourru, and Ravi 2018), the focus of our work is not on generating a clustering; instead, the goal is to explain the results of clustering algorithms. While the cluster description problem considered here uses a formulation similar to the one used in (Davidson, Gourru, and Ravi 2018), their focus was on expressing the problem as an ILP and solving it optimally using public domain ILP solvers. In particular, approximation algorithms with provable performance guarantees were not considered in (Davidson, Gourru, and Ravi 2018).

Additional details. Due to the space limits, several proofs are omitted; they can be found in (Sambaturu et al. 2019).

2 Preliminaries

Notation and Definitions. Let $S = \{s_1, \ldots, s_n\}$ be a set of n objects, and $\pi = \{C_1, \ldots, C_k\}$ be a partition of S into $k \geq 2$ clusters. Let T be the universe of m tags such that each object $s_i \in S$ is associated with a subset $t_i \subseteq T$ of tags. A solution is a subset $X \subseteq T$, and will be represented as a partition $X = (X_1, \ldots, X_k)$, where X_ℓ is the descriptor (i.e., subset of tags) used for cluster C_ℓ. We say that $s_i \in S$ is *covered* by a set $X \subseteq T$ of tags if $X \cap t_i \neq \emptyset$. Let $E(j) = \{s_i : j \in t_i\}$ be the set of all objects that can be covered by the tag $j \in T$. Let $\eta = \max_j |E(j)|$ denote the maximum number of objects covered by any tag in T. Let $\gamma = \max_i |t_i|$ denote the maximum number of tags associated with any object in S. Objects $s_i, s_{i'} \in S$ are said to be *dependent* if $t_i \cap t_{i'} \neq \emptyset$, i.e., if their tag sets overlap. Let $\Delta(i) = |\{i' : t_i \cap t_{i'} \neq \emptyset\}|$ denote the degree of dependence of s_i, and let $\Delta = \max_i \Delta(i)$ be the maximum dependence. Finally, for a solution $X = (X_1, \ldots, X_k)$, let $V_\ell(X) = \{s_i \in C_\ell : t_i \cap X_\ell \neq \phi\}$ be the subset of objects in C_ℓ covered by X, $1 \leq \ell \leq k$. For any integer $k \geq 1$, we use $[k]$ to denote the set $\{1, \ldots, k\}$.

Problem statement. Our objective is to find a solution X that *simultaneously* ensures high coverage $|V_\ell(X)|$ in each cluster C_ℓ, $1 \leq \ell \leq k$. An obvious choice is to consider a max-min type of objective $X = \text{argmax} \min_\ell |V_\ell(X)|$ (see, e.g., (Udwani 2017)). However, this doesn't allow domain specific coverage requirements (e.g., higher coverage for the cluster of threat sequences in genomic data). Therefore, we consider a more general formulation, which specifies a coverage requirement for each cluster.

Minimum Constrained Cluster Description (MinConCD)

Instance: A set $S = \{s_1, \ldots, s_n\}$ of objects, a partition $\pi = \{C_1, \ldots, C_k\}$ into $k \geq 2$ clusters, a universe T of m tags, tag set $t_i \subseteq T$ for each object s_i and parameter M_ℓ for each cluster C_ℓ, $1 \leq \ell \leq k$.

Requirement: Find a solution $X = (X_1, \ldots, X_k)$ that minimizes the cost $\sum_{\ell=1}^{k} |X_\ell|$ and satisfies the following constraints: (i) the subsets in X are pairwise-disjoint and (ii) for each cluster C_ℓ, $|V_\ell(X)| \geq M_\ell$.

Comparison with the formulation in (Davidson, Gourru, and Ravi 2018). The main problem considered in (Davidson, Gourru, and Ravi 2018) is to minimize the cost (i.e., $\sum_\ell |X_\ell|$) under the constraint that all the objects in S are covered. This was formulated as an integer linear program (ILP). That reference also presented an ILP for minimizing the cost under the requirement that at least a total of $\alpha|S|$ objects are covered over all the clusters for a given α, $0 < \alpha \leq 1$. (This was called the "cover or forget" formulation in (Davidson, Gourru, and Ravi 2018).) One difficulty with this formulation is that solutions that satisfy the total coverage requirement may cover a large percentage of the objects in some clusters while covering only a small percentage of those in other clusters. (We will present an example of this phenomenon using a real life dataset in Section 5.) Our formulation avoids this difficulty by allowing the specification of the coverage requirement for each cluster separately. As mentioned in Section 1, our formulation in conjunction with the disjointness requirement introduces additional challenges in developing efficient approximation algorithms with provable performance guarantees.

Approximation algorithms. As shown in (Sambaturu et al. 2019), if the coverage requirements must be met, then the cost cannot be approximated to within any factor $\rho \geq 1$, unless **P = NP**. Therefore, we study bi-criteria approximation algorithms. We say that a solution X is an (α, δ)-approximation if (i) for each cluster C_ℓ, $|V_\ell(X)| \geq \alpha M_\ell$ and (ii) $\sum_\ell |X_\ell| \leq \delta B^*$, where B^* is the optimal cost.

Other variations. Another variation we will explore, referred to as MINCONCDO, allows limited overlap between different descriptors. Given input parameters M_ℓ for $\ell \in [k]$ and overlap limit B_o, the objective here is to find a solution $X = (X_1, \ldots, X_k)$ of minimum cost such that $|V_\ell(X)| \geq M_\ell$ for each ℓ and $\sum_{\ell \neq \ell'} |X_\ell \cap X_{\ell'}| \leq B_o$; this problem is discussed in (Sambaturu et al. 2019).

3 Algorithm ROUND: approximation using Linear Programming and Rounding

Our approach for approximating MinConCD is based on LP relaxation and then rounding the fractional solution. This is a common approach for many combinatorial optimization problems, especially those with covering constraints (see, e.g., (Williamson and Shmoys 2011)). However, the disjointness requirement for descriptors poses a challenge in terms of dependencies and requires a new method of rounding. We start with an ILP formulation.

ILP Formulation. For each $j \in T$ and $\ell \in [k]$, $x_\ell(j)$ is an indicator (i.e., $\{0,1\}$-valued) variable, which is 1 if tag $j \in X_\ell$. We have an indicator variable $z(i)$ for each $s_i \in S$, which is 1 if object s_i is covered. The objective and constraints of the ILP formulation are as follows.

$$(\mathcal{IP}) \qquad \text{Minimize} \sum_{\ell=1}^{k} \sum_{j \in T} x_\ell(j) \ \text{ such that}$$

$$\forall \ell, \ \forall s_i \in C_\ell : \sum_{j \in t_i} x_\ell(j) \ \geq \ z(i), \ \ \forall \ell : \sum_{s_i \in C_\ell} z(i) \ \geq \ M_\ell$$

$$\forall j : \sum_{\ell} x_\ell(j) \ \leq \ 1, \ \ \text{All variables} \in \{0, 1\}$$

Algorithm 1 describes the steps of ROUND. The linear program $\mathcal{P}$ (from Step 1) can be solved (Step 2) using standard techniques in polynomial time (e.g., (Kleinberg and Tardos 2006)) and a fractional solution to the variables of $\mathcal{P}$ can be obtained efficiently whenever there is a feasible solution. We analyze the performance of ROUND in Theorem 1. Most of our discussion will focus on analyzing the solution $X = (X_1, \ldots, X_k)$ computed in any iteration of Steps 4–14 of Algorithm 1. For each ℓ, define $Z_\ell = \sum_{s_i \in C_\ell} Z(i)$. A proof of the following lemma appears in (Sambaturu et al. 2019).

Lemma 1. *For each $\ell \in [k]$, the expected number of objects covered in cluster C_ℓ by a solution X in any round of Step 4 of algorithm* ROUND *is at least $M_\ell/4$.*

Challenge in deriving a lower bound on the number of objects covered in each cluster. Lemma 1 implies that, in

Algorithm 1: Algorithm ROUND

Input : $S, \pi = \{C_1, \ldots, C_k\}, T, M_\ell$ for each $\ell = 1, \ldots, k$. (Note: $|S| = n$.)
Output: $X = (X_1, \ldots, X_k)$

Let $\mathcal{P}$ be a linear relaxation of the ILP $\mathcal{IP}$, obtained by requiring all variables to be in $[0, 1]$ (instead of being binary).
Solve $\mathcal{P}$. If it is not feasible, return "no feasible solution". Else, let x^*, z^* denote the optimal fractional solution and B denote the associated cost.
For all j, and for all ℓ, set $x_\ell(j) = x^*_\ell(j)/2$, and for all s_i set $z(i) = z^*(i)/2$.
for $4 \ln n$ *times* **do**
  for $j \in T$ *and* $\ell = 1, \ldots, k$ **do**
    With probability $x_\ell(j)$, round $X_\ell(j) = 1$ and $X_{\ell'}(j) = 0$ for all $\ell' \neq \ell$.
    With probability $1 - \sum_\ell x_\ell(j)$, set $X_{\ell'}(j) = 0$ for all ℓ'.
  end
  for $s_i \in S$ **do**
    If $X_\ell(j) = 1$ for some $j \in t_i$, define $Z(i) = 1$.
  end
  For each ℓ, define $Z_\ell = \sum_{s_i \in C_\ell} Z(i)$.
  If $Z_\ell \geq M_\ell/8$ for each ℓ, and $\sum_\ell \sum_j X_\ell(j) \leq 2B$, return X as the solution and **stop**.
end
Return failure.

expectation, a constant fraction of the objects in each cluster are covered. If the variables $Z(i)$ were all independent, we could use a Chernoff bound (see, e.g., (Dubhashi and Panconesi 2009)) to show that Z_ℓ is concentrated around $E[Z_\ell] = \Theta(M_\ell)$. However, $Z(i)$ and $Z(i')$ are dependent when $t_i \cap t_{i'} \neq \emptyset$. Therefore, the standard Chernoff bound cannot be used in this case, and a new approach is needed. We address this issue by considering the number of objects which are *not covered* in C_ℓ. There are dependencies in this case as well; however, they are of a special type, which can be handled by Janson's upper tail bound (Janson and Rucinski 2002), which is one of the few known concentration bounds for dependent events.

For $s_i \in C_\ell$, let $Y(i) = 1 - Z(i)$ be an indicator for s_i not covered by the solution X. We apply the upper tail bound of (Janson and Rucinski 2002) to obtain a concentration bound on $Y_\ell = \sum_{s_i \in C_\ell} Y(i)$. We first observe that the dependencies among the Y_ℓ variables are of the form considered in (Janson and Rucinski 2002). Let $\Gamma = T$, and let $\xi_{j\ell}$ be an indicator that is 1 if $X_\ell(j) = 0$. For a fixed ℓ, the variables $X_\ell(j)$ are independent over all j, since ROUND rounds the variables for each j independently. Hence, for a fixed ℓ, the random variables $\xi_{j\ell}$ are all independent. Then, for $s_i \in C_\ell$, $Y(i) = \prod_{j \in t_i} \xi_{j\ell}$. This implies that $Y(i)$ and $Y(i')$ are independent if $t_i \cap t'_i = \emptyset$. Therefore, the random variables $Y(i)$ are of the type considered in (Janson and Rucinski 2002). Let Δ be the maximum number of sets $t_{i'}$ which intersect with any t_i, as defined earlier in Section 2, and let $\lambda = E[Y_\ell]$. Then, the bound from (Janson and Rucinski 2002) gives

$$\Pr[Y_\ell \geq \lambda + t] \leq (\Delta + 1) \exp\Big(- \frac{t^2}{4(\Delta + 1)(\lambda + t/3)} \Big).$$

We use this bound in the lemma below.

Lemma 2. *Assume $M_\ell \geq a|C_\ell|$ for all $\ell \in [k]$ for a constant $a \in (0, 1]$, and let $(\Delta+1) \leq \min_{\ell \in [k]} \frac{d|C_\ell|}{\log n}$, where $d \leq \frac{a^2}{576}$. Let $t = M_\ell/8$. Then, for any fixed $\ell \in [k]$ and any round of Step 4 of* ROUND, $\Pr(Y_\ell \geq E[Y_\ell] + t) \leq \frac{1}{n}$.

Proof. We consider a fixed ℓ here. We have $\Pr[Y(i) = 1] = 1 - \Pr[Z(i) = 1] \leq 1 - z^*(i)/2$, from the proof of Lemma 1. Next, $\sum_{s_i \in C_\ell} Y(i) + Z(i) = Y_\ell + Z_\ell = |C_\ell|$. Therefore, from Lemma 1

$$E[Y_\ell] = |C_\ell| - E[Z_\ell] \leq |C_\ell| - \frac{M_\ell}{4} \leq (1 - \frac{a}{4})|C_\ell|,$$

since $M_\ell \geq a|C_\ell|$. Let $\lambda = E[Y_\ell]$. Then

$$(\Delta + 1)\lambda \leq \frac{d|C_\ell|^2(1 - \frac{a}{4})}{\log n} \leq \frac{d|C_\ell|^2(1 - \frac{a}{4})a^2}{a^2 \log n} \leq \frac{d(1 - \frac{a}{4})64t^2}{a^2 \log n}$$

Also

$$(\Delta + 1)\frac{t}{3} \leq \frac{d|C_1|ta}{3a \log n} \leq \frac{dM_1 t}{3a \log n} = \frac{8dt^2}{3a \log n}$$

Therefore,

$$4(\Delta + 1)(\lambda + \frac{t}{3}) \leq \frac{8dt^2}{a \log n}\Big[\frac{8(1 - a/4)}{a} + \frac{1}{3}\Big]$$

Putting these together, we have

$$\frac{t^2}{4(\Delta + 1)(\lambda + \frac{t}{3})} \geq \frac{\log n}{\frac{8d}{a}[\frac{8(1-a/4)}{a} + \frac{1}{3}]} \geq 2 \log n,$$

where the last inequality follows because $d \leq \frac{a^2}{576}$; thus, $\frac{8d}{a}\Big[\frac{8(1-a/4)}{a} + \frac{1}{3}\Big] \leq \frac{1}{2}$. Applying Janson's upper tail bound,

$$\begin{aligned} \Pr(Y_\ell \geq \lambda + t) &\leq (\Delta + 1)\exp\Big(\frac{-t^2}{4(\Delta + 1)(\lambda + t/3)}\Big) \\ &\leq (\Delta + 1)\exp(-2 \log n) \leq \frac{(\Delta + 1)}{n^2} \\ &\leq \frac{M_\ell}{n^2} \leq \frac{1}{n}, \end{aligned}$$

where the last inequality is because $M_\ell \leq |C_\ell| \leq n$. □

Theorem 1. *Suppose an instance of MinConCD satisfies the following conditions: (1) $M_\ell \geq a|C_\ell|$ for all $\ell \in [k]$, and for some constant $a \in (0, 1]$, and (2) $(\Delta + 1) \leq \min_\ell \frac{d|C_\ell|}{\log n}$ and $d \leq \frac{a^2}{576}$, and (3) $k \leq n/4$. If the LP relaxation $(\mathcal{P})$ is feasible, then with probability at least $1 - \frac{1}{n}$, algorithm* ROUND *successfully returns a solution X, which is an $(1/8, 2)$–approximation.*

Proof. We analyze the properties of a solution X computed in each round of Step 4 of ROUND. From Lemma 2, for any $\ell \in [k]$, $Y_\ell \leq E[Y_\ell] + \frac{M_\ell}{8}$ with probability at least $1 - \frac{1}{n}$. Substituting $E[Y_\ell] \leq |C_\ell| - \frac{M_\ell}{4}$ (shown in the proof of

Lemma 2) in the above equation we have,

$$Y_\ell \leq |C_\ell| - \frac{M_\ell}{4} + \frac{M_\ell}{8} \leq |C_\ell| - \frac{M_\ell}{8}$$

with the same probability, for each ℓ. Therefore,

$$Z_\ell \geq |C_\ell| - Y_\ell \geq |C_\ell| - \Big(|C_\ell| - \frac{M_\ell}{8}\Big) \geq \frac{M_\ell}{8},$$

for each ℓ, with probability at least $1 - 1/n$. This implies $\Pr[Z_\ell < \frac{M_\ell}{8}] \leq 1/n$, so that $\Pr[\exists\ \ell \text{ with } Z_\ell < \frac{M_\ell}{8}] \leq k/n$. Therefore, with probability at least $1 - k/n$, for all $\ell \in [k]$, we have $Z_\ell \geq \frac{M_\ell}{8}$.

Next, we consider the cost of the solution (i.e., the total number of tags used). The rounding ensures that $\Pr[X_\ell(j) = 1] = x_\ell(j)$, for each ℓ, j. Thus, by linearity of expectation, the expected cost of the solution is

$$E\Big[\sum_\ell \sum_j X_\ell(j)\Big] = \sum_j \sum_\ell x_\ell(j) \leq B$$

By Markov's inequality, $Pr[\sum_\ell \sum_j X_\ell(j) > 2B] \leq \frac{1}{2}$.

Putting everything together, for each round, the probability of success (i.e., the cost is at most $2B$ and $Z_\ell \geq M_\ell/8$ for each ℓ) is at least $\frac{1}{2} - \frac{k}{n} \geq \frac{1}{4}$, since $k \leq n/4$. Therefore, the probability that at least one of the $4 \ln n$ rounds is a success is at least $1 - (\frac{3}{4})^{4 \ln n} \geq 1 - \frac{1}{n}$. □

4 Approximation using submodularity

The MinConCD problem can be viewed as a problem of submodular function maximization with constraints which can be expressed as a matroid. For convenience, we assume that a cost budget B is also specified as part of an instance of MinConCD and that the goal of the (α, δ)–approximation algorithm is to produce a solution that covers at least αM_ℓ objects in each cluster C_ℓ and has a cost of at most δB. This assumption can be made without loss of generality since the optimal cost is an integer in $[1 \,..\, |T|]$; one can do a binary search over this range by executing the algorithm with $O(\log |T|)$ different budget values and using the smallest budget for which the algorithm produces a solution. We first discuss the necessary concepts, and then describe our algorithm. We refer the reader to (Calinescu et al. 2011) for more details regarding submodular function maximization subject to matroid constraints.

A matroid is a pair $\mathcal{M} = (Y, \mathcal{I})$, where $\mathcal{I} \subseteq 2^Y$ and (1) $\forall A' \in \mathcal{I}$, $A \subset A' \Rightarrow A \in \mathcal{I}$, and (2) $\forall A, A' \in \mathcal{I}$, $|A| < |A'| \Rightarrow \exists x \in A' - A$ such that $A \cup \{x\} \in \mathcal{I}$. A function $f : 2^Y \to \mathbb{R}_{\geq 0}$ is submodular if $f(A \cup \{x\}) - f(A) \geq f(A' \cup \{x\}) - f(A')$ for all $A \subseteq A'$. Function $f(\cdot)$ is monotone if $f(A) \leq f(A')$ for all $A \subseteq A'$.

Constructing a matroid for MinConCD. For each tag $j \in T$, let $Y_j = \{a_j, b_j\}$. Let $Y = \cup_j Y_j$. Let $\mathcal{I} = \{A \subset Y : |A \cap Y_j| \leq 1,\ \forall j \text{ and } |A| \leq B\}$. Then $\mathcal{M} = (Y, \mathcal{I})$ can be seen as an intersection of a partition matroid, which requires $|A \cap Y_j| \leq 1$ for all j, and a uniform matroid, which requires $|A| \leq B$.

Lemma 3. *$\mathcal{M} = (Y, \mathcal{I})$ is a matroid.*

Constructing a submodular function. It is easy to verify that the function $|V_\ell(X)|/M_\ell$ (which is the fraction of objects covered by solution X in cluster C_ℓ) is a submodular function of X. When $k = 2$, we need to find a solution X such that $|V_1(X)|/M_1 \geq 1$ and $|V_2(X)|/M_2 \geq 1$ hold *simultaneously*. This can be achieved by requiring $\min\Big\{\frac{|V_1(X^A)|}{M_1}, \frac{|V_2(X^A)|}{M_2}\Big\} \geq 1$. However, the minimum of two submodular functions is not submodular, in general. We handle this by using the "saturation" technique of (Krause, McMahan, and Guestrin 2008): for $A \subseteq Y$, define $X_1^A = \{j \in T : a_j \in A\}$ and $X_2^A = \{j \in T : b_j \in A\}$, and let $X^A = (X_1^A, X_2^A)$. Define $F_1(A) = \min\Big\{\frac{|V_1(X^A)|}{M_1}, 1\Big\}$, $F_2(A) = \min\{\frac{|V_2(X^A)|}{M_2}, 1\}$ and $F(A) = F_1(A) + F_2(A)$. It is easy to verify the following lemma.

Lemma 4. *$F_1(A)$, $F_2(A)$, and $F(A)$ are monotone submodular functions of A.*

Our algorithm for MinConCD with $k = 2$ involves the following steps.

1. Use the algorithm of (Calinescu et al. 2011) to find a set $A \in \mathcal{I}$ which maximizes $F(A) = F_1(A) + F_2(A)$.
2. Return the solution $X^A = (X_1^A, X_2^A)$.

Theorem 2. *Suppose there is a feasible solution to an instance (S, π, T, B, M_1, M_2) of MinConCD, with $k = 2$. Then, the above algorithm runs in polynomial time and returns an $(1 - 2/e, 1)$-approximate solution X^A.*

In (Sambaturu et al. 2019), we show how the above approach can be extended to approximate the objective of maximizing the total coverage (i.e., $\sum_\ell |V_\ell(X)|$), for any k.

5 Experimental results

Our experiments focus on the following questions.

1. Benefit of allowing cluster specific coverage. How do the results from our MinConCD formulation compare with those of (Davidson, Gourru, and Ravi 2018)?

2. Dependence of the cost on coverage. Does the cost increase gradually as the coverage requirement increases?

3. Descriptions with pairs of tags. Does adding pair of tags to the tagset provide better explanations of clusters? How do these results compare to those where pairs of tags are not used?

4. Performance. Does ROUND give solutions with good approximation guarantees in practice, and does it scale to large real world datasets?

5. Explanation of clusters. Do the solutions provide interpretable explanations of clusters in real world datasets?

5.1 Datasets and methods

Datasets. Table 1 provides details of the real and synthetic datasets used in our experiments. In the synthetic datasets, an object is associated with a tag with probability p.

The Threat and Uniref90 datasets (Jain and Kihara 2018; Ramesh and Warren 2018) contain genome sequences and information that may indicate a given gene's threat potential, which is established manually by domain experts— this is used to partition the sequences into four clusters (referred to as threat bins 1–4). The tags associated with these sequences are various characteristics of the genes in them, obtained from Bioinformatics repositories. Uniref90 is an expanded version of the Threat dataset, with additional attributes computed using sequence similarity.

Real/Synthetic	$\|S\|$	$\|T\|$	$\|C_1\|$	$\|C_2\|$
Genome (Threat)	248	4632	73	175
Uniref90	21537	2193	13406	8131
Flickr	2455	175	1052	1402
Philosophers	249	14549	110	139
Synthetic-1	100	100	48	52
Synthetic-2	1000	1000	502	498
Synthetic-3	1000	1000	478	522

Table 1: Description of datasets. The three synthetic datasets above were generated using probability values 0.05, 0.2 and 0.05 respectively.

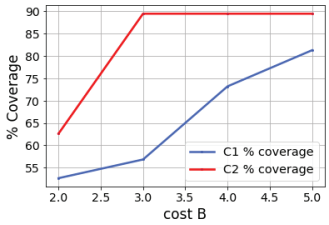
The Flickr dataset (Yang, McAuley, and Leskovec 2013) consists of images as nodes and relationships between images as edges. A relationship could correspond to images being submitted from the same location, belonging to the same group, or sharing common tags, etc. We use the Louvain algorithm in Networkx (Aric A. Hagberg and Swart 2008) to generate communities of images, and pick the communities as clusters. User defined tags, such as "dog", "person", "car", etc., are provided for each image. The Philosophers dataset (Yang, McAuley, and Leskovec 2013) consists of Wikipedia articles (considered to be the objects to be clustered) on various philosophers. The tags corresponding to each object are the non-philosopher Wikipedia articles to which there is an outlink from the philosopher article. The clusters in the philosopher data are generated by grouping communities that share a common keyword as a single cluster. The Synthetic-2 and Synthetic-3 datasets are generated with four clusters. In some experiments, we merge the clusters in these datasets into two clusters, one corresponding to clusters 1 and 2, and the other corresponding to clusters 3 and 4, as shown in Table 1. In many of our experiments, we consider $k = 2$, and fixed M_1 and M_2 close to 70% of that of $|C_1|$ and $|C_2|$, respectively. The Twitter dataset used in (Davidson, Gourru, and Ravi 2018) was unavailable due to the terms of the dataset, and we are unable to compare with the results of (Davidson, Gourru, and Ravi 2018).

Methods. We study the performance of ROUND in our experiments. We run the rounding steps 4-14 in Algorithm 1 for 10 iterations. We use the ILP as a baseline. Note that for the complete coverage version (i.e., $M_\ell = |C_\ell|$), the ILP is exactly the method used by Davidson et al. (2018).

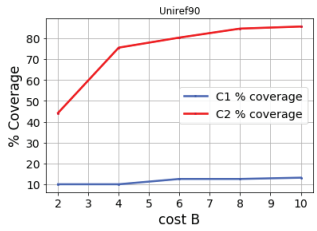
Code. The code is available at https://github.com/prathyush6/ExplainabilityCodeAAAI20.git.

5.2 Results

1. Benefit of allowing cluster specific coverage. The exact coverage formulation of (Davidson, Gourru, and Ravi 2018) (which corresponds to $M_\ell = |C_\ell|$ for all ℓ) is infeasible for some of the datasets we consider. Instead, we examine the cluster descriptions computed using the cover-or-forget formulation of (Davidson, Gourru, and Ravi 2018), which maximizes the total number of objects covered. Figures 1(a) and 1(b) show the coverage percent for each cluster, i.e., $(|V_\ell(X)|/M_\ell) \times 100\%$, ($y$-axis) versus the cost of the solution (x-axis), for the Flickr and Uniref90 datasets, respectively. Both figures show that the coverage is highly imbalanced. For instance, with 3 tags, almost 90% of elements in cluster C2 are covered, whereas only 57% of elements in C1 are covered in Figure 1(a). This is a limitation of the cover-or-forget approach, and the cluster specific coverage requirements in MinConCD can help alleviate this problem.



(a) Flickr



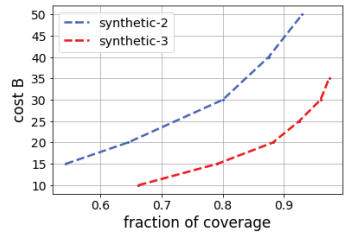
(b) Uniref90

Figure 1: Coverage percent in each cluster (y-axis) and the solution cost (x-axis) for the Flickr and Uniref datasets.

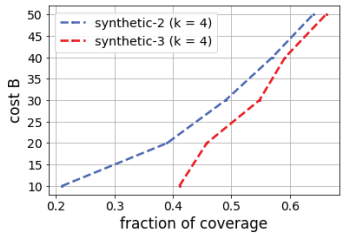
2. Dependence of cost on the coverage requirement. Figures 2(a) and 2(b) show the cost of the solution vs the coverage fraction. Initially, the cost grows slowly, but after a point, the cost increases rapidly. For some parameter settings, there is no feasible solution, which corresponds to the ends of the curves. As the number of clusters increases, the cost to cover a given fraction of elements increases.

3. Descriptions with pairs of tags. We extend the set of tags T to T_{ext} by adding every pair (j, j'), where $j, j' \in T$, and use T_{ext} for finding descriptions. For some datasets, this increases the feasible regime, but when the instance is feasible, the solutions using T and T_{ext} are pretty close. However, even if the description cost is very similar, using T_{ext} sometimes provides more meaningful descriptions. For instance, on Philosophers dataset, we found pairs such as ('Benedict_XIV', 'Roman_Catholic_religious_order') picked to describe the cluster corresponding to Wikipedia articles related to Christianity.

4. Performance. First, we consider the approximation guarantee of ROUND in practice. Figure 3(a) shows the approximation ratios (i.e., the ratio of the coverage achieved by ROUND, to that of an optimum solution) on the y-axis, and the solution cost on the x-axis. Recall that the analysis in Theorem 1 only guarantees a coverage factor of $1/8 = 0.125$, but the plot for $k = 2$ shows that the approximation factors in practice are much higher—they are always ≥ 0.8, and > 0.9 in most cases. Note that the curves are non-monotone—this is due to the stochastic nature of ROUND. However, for $k = 4$, the approximation ratios are lower as shown in Figure 3(b).

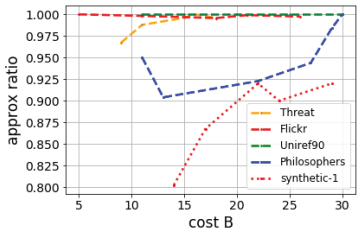


(a) $k = 2$ clusters

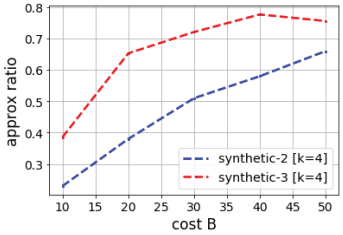


(b) $k = 4$ clusters

Figure 2: Overall fraction of coverage (x-axis) vs the cost B (y-axis). The minimum coverage requirement in each cluster is set to at least 50%.



(a) Datasets with 2 clusters



(b) Datasets with 4 clusters

Figure 3: Approximation ratio of ROUND (y-axis) vs budget (x-axis) for different datasets (higher is better).

We also observe that ROUND is quite scalable. The running time is dominated by the time needed to solve the LP. We use the Gurobi solver, which is able to run successfully on datasets whose data matrix (i.e., the matrix of objects and tags) has up to 10^8 entries. In contrast, the ILP does not scale beyond datasets with more than 10^6 entries.

5. Explanation of clusters

(a) Genomic threat sequences (harmful v harmless). Our method chose 13 tags for the harmful cluster. Upon expert review of our results, we found that certain tags served as indicators that genes found within the harmful cluster can intrinsically be viewed as harmful, while others may need to act in concert, be viewed in combination with other tags, or be representative of selection bias. Of the 13 tags selected, 4 indicate intrinsic capability of being harmful: KW-0800 (toxin), 155864.Z3344 (Shiga toxin 1), IPR011050 (Pectin lyase fold/virulence), and IPR015217 (invasin domain). Another 4 tags are suggestive that the genes implicated are involved in processes or locations commonly associated with threat: KW-0732 (signal peptide), KW-0614 (plasmid), KW-0964 (secreted), and GO:0050896 (response to stimulus). Other tags associated with the threat partition such as KW-0002 (3-D structure) indicate a limited amount of data and perhaps bias in the research literature for the clusters analyzed. The definitions of these tags are presented in (Sambaturu et al. 2019).

To define the clusters, each gene was used as a seed to obtain constituent members of Uniref90 groups. By including genes greater than or equal to 90% sequence identity to the manually curated set, clusters were created and the number of sequences with associated attributes increased to 63,305.

(b) Philosophers dataset: Here, Cluster 1 is the set of Wikipedia pages related to India and Greece, whereas Cluster 2 has pages related to Christianity. The tags picked by our algorithm to explain Cluster 1 are *Buddhist terms and concepts, Metaphysics, Sanyasa, Buddhism, Athenian, Mathematician, Greek Language*, which are consistent with the pages in the cluster. The tags picked to explain Cluster 2 are *Constantinople, Existentialism, Abortion, Political Philosophy, Theology, England*, which are consistent with the contents of that cluster.

6 Conclusions

We formulated a version of the cluster description problem that allows simultaneous coverage requirements for all the clusters. We presented rigorous approximation algorithms for the problem using techniques from randomized rounding of linear programs and submodular optimization. Our rounding-based algorithm exhibits very good performance in practice. Using a real world data set containing genomic threat sequences, we observed that the descriptors found using the algorithm give useful insights. In our experiments, we considered several different parameters including coverage level, budget, and overlap. Using these parameters, our approach can be used to obtain a range of solutions from which a practitioner can choose appropriate descriptors.

Acknowledgments. We thank the reviewers for carefully reading the manuscript and providing very helpful comments. This work has been partially supported by DTRA CNIMS (Contract HDTRA1-11-D-0016-0001), NSF Grant IIS-1908530, NSF Grant IIS-1910306, NSF Grant OAC-1916805, NSF CRISP 2.0 Grant 1832587, NSF DIBBS Grant ACI-1443054, NSF BIG DATA Grant IIS-1633028 and NSF EAGER Grant CMMI-1745207. The research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Army Research Office (ARO) under cooperative Agreement Number W911NF-17-2-0105. The views and conclusions con-

tained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, ARO, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

Aric A. Hagberg, D. A. S., and Swart, P. J. 2008. Exploring network structure, dynamics, and function using networkx. In *7th Python in Science Conference (SciPy2008), (Pasadena, CA USA)*, 11–15.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):P10008.

Bolla, M. 2013. *Spectral clustering and biclustering: learning large graphs and contingency tables*. Hoboken, NJ: Wiley.

Calinescu, G.; Chekuri, C.; Pál, M.; and Vondrák, J. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.* 40(6):1740–1766.

Dang, X. H., and Bailey, J. 2010. Generation of alternative clusterings using the CAMI approach. In *SDM*, volume 10, 118–129. SIAM.

Dao, T.; Kuo, C.; Ravi, S. S.; Vrain, C.; and Davidson, I. 2018. Descriptive clustering: ILP and CP formulations with applications. In *Proc. IJCAI*, 1263–1269.

Davidson, I.; Gourru, A.; and Ravi, S. S. 2018. The cluster description problem – Complexity results, formulations and approximations. In Proc. NeurIPS, 6193–6203.

Dubhashi, D. P., and Panconesi, A. 2009. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press.

Fortunato, S. 2010. Community detection in graphs. *Physics Rep.* 486(3-5):75–174.

Goodman, B., and Flaxman, S. 2016. EU regulations on algorithmic decision-making and a "right to explanation". Presented at 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016), New York, NY.

Gunning, D. 2017. Explainable Artificial Intelligence (XAI). DARPA Program Update Document. Available from https://www.darpa.mil/attachments/XAIProgramUpdate.pdf.

Hamid, M. N., and Friedberg, I. 2018. Identifying antimicrobial peptides using word embedding with deep recurrent neural networks. Bioinformatics (submitted). Preprint: Biorxiv https://doi.org/10.1101/255505.

Han, J., and Kamber, M. 2011. *Data Mining: Concepts and Techniques*. San Mateo, CA: Morgan Kauffmann Publishers.

Jain, A., and Kihara, D. 2018. Phylo-PFP: Improved automated protein function prediction using phylogenetic distance of distantly related sequences. Bioinformatics (to appear).

Jain, A.; Gali, H.; and Kihara, D. 2018. Identification of moonlighting proteins in genomes using text mining techniques. PROTEOMICS: https://onlinelibrary.wiley.com/doi/abs/10.1002/pmic.201800083.

Janson, S., and Rucinski, A. 2002. The infamous upper tail. *Random Struct. Algorithms* 20:317–342.

Khuller, S.; Moss, A.; and Naor, J. 1999. The budgeted maximum coverage problem. *Inf. Proc. Lett.* 70(1):39–45.

Kleinberg, J., and Tardos, E. 2006. *Algorithm Design*. New York, NY: Pearson Publishing.

Krause, A.; McMahan, H. B.; and Guestrin, C. 2008. Robust submodular observation selection. *Journal of Machine Learning Research (JMLR)* 2761–2801.

Kuo, C.; Ravi, S. S.; Dao, T.; Vrain, C.; and Davidson, I. 2017. A framework for minimal clustering modification via constraint programming. In *Proc. AAAI*, 1389–1395.

2017. Proc. IJCAI-17 Workshop on Explainable AI (XAI). http://www.intelligentrobots.org/files/IJCAI2017/IJCAI-17_XAI_WS_Proceedings.pdf.

2018. Proc. IJCAI-ECAI-2018 Workshop on Explainable AI (XAI). https://www.dropbox.com/s/jgzkfws41ulkzxl/proceedings.pdf?dl=0.

Qi, Z., and Davidson, I. 2009. A principled and flexible framework for finding alternative clusterings. In *Proc. 15th ACM SIGKDD*, 717–726.

Ramesh, S., and Warren, A. 2018. The learnability of taxonomic divisions. Talk and poster presentation at ISMB.

Sambaturu, P.; Gupta, A.; Davidson, I.; Ravi, S. S.; Vullikanti, A.; and Warren, A. 2019. Full Version: Efficient Algorithms for Generating Provably Near-Optimal Cluster Descriptors for Explainability. Technical Report. https://tinyurl.com/6094supplement.

Srinivasan, A. 1995. Improved approximations of packing and covering problems. In *Proc. STOC*, 268–276.

Tan, P.; Steinbach, M.; and Kumar, V. 2006. *Data Mining*. New York, NY: Pearson Publishing Co.

Udwani, R. 2017. Multi-objective maximization of monotone submodular functions with cardinality constraint. *CoRR* abs/1711.06428.

Williamson, D. P., and Shmoys, D. B. 2011. *The Design of Approximation Algorithms*. Cambridge University Press.

Yang, J.; McAuley, J. J.; and Leskovec, J. 2013. Community detection in networks with node attributes. In *Proc. ICDM*, 1151–1156.

Zaki, M., and Meira, W. 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. New York, NY: Cambridge University Press.