A Generic Framework for Interesting Subspace Cluster Detection in Multi-attributed Networks

Feng Chen, Baojian Zhou *, and Adil Alim *
Computer Science Department
University at Albany – SUNY
fchen5,bzhou6,aalimu@albany.edu

Liang Zhao
Department of Information Science and Technology
George Mason University
lzhao9@gmu.edu

***Abstract*—Detection of interesting (e.g., coherent or anomalous) clusters has been studied extensively on plain or univariate networks, with various applications. Recently, algorithms have been extended to networks with multiple attributes for each node in the real-world. In a multi-attributed network, often, a cluster of nodes is only interesting for a subset (subspace) of attributes, and this type of clusters is called *subspace clusters*. However, in the current literature, few methods are capable of detecting subspace clusters, which involves concurrent feature selection and network cluster detection. These relevant methods are mostly heuristic-driven and customized for specific application scenarios.**

In this work, we present a generic and theoretical framework for detection of interesting subspace clusters in large multi-attributed networks. Specifically, we propose a subspace graph-structured matching pursuit algorithm, namely, `SG-Pursuit`, to address a broad class of such problems for different score functions (e.g., coherence or anomalous functions) and topology constraints (e.g., connected subgraphs and dense subgraphs). We prove that our algorithm 1) runs in nearly-linear time on the network size and the total number of attributes and 2) enjoys rigorous guarantees (geometrical convergence rate and tight error bound) analogous to those of the state-of-the-art algorithms for sparse feature selection problems and subgraph detection problems. As a case study, we specialize `SG-Pursuit` to optimize a number of well-known score functions for two typical tasks, including detection of coherent dense and anomalous connected subspace clusters in real-world networks. Empirical evidence demonstrates that our proposed generic algorithm `SG-Pursuit` is superior over state-of-the-art methods that are designed specifically for these two tasks.

I. Introduction

With recent advances in hardware and software technologies, the huge volumes of data now being collected from multiple sources are naturally modeled as *multi-attributed networks*. For example, massive multi-attributed biological networks have been created by integrating gene expression data with secondary data such as pathway or protein-protein interaction data for improved outcome prediction of cancer patients [23]. Other examples include the multi-attributed networks that combine "Big data" (e.g., Twitter feeds) and traditional surveillance data for influenza studies [10] and the social networks that contain both the friendship relations and user attributes such as interests, frequencies of keywords mentioned in posts, and demographics [16].

As one of the major tasks in network mining, the detection of interesting clusters in attributed networks, such as coherent or anomalous clusters, has attracted a great deal of attention in many applications, including medicine and public health [16], [25], law enforcement [35], cyber security [28], among others [32], [33], [35]. To deal with the multiple or even high-dimensional attributes, most existing methods

* These two authors contributed equally.

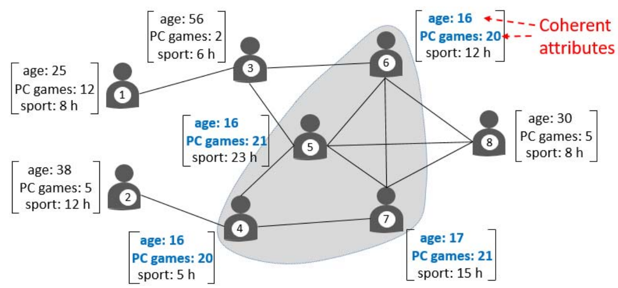

Fig. 1: A social network with three attributes (age, PC games, and sport) in each user node and one potential coherent dense subspace cluster (highlighted in the shaded region and blue-colored texts) that has a coherent subset of attributes (age and PC games) and a dense subgraph of nodes (4, 5, 6, and 7). This cluster might be of interest to video game producers. (Adapted from [16])

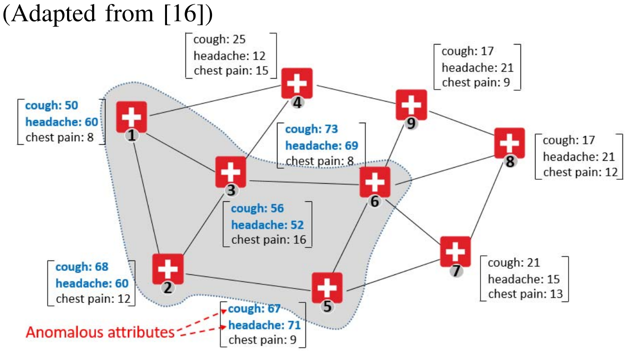

Fig. 2: A health surveillance network of emergency departments (EDs) with three attributes (counts of cases of three different ICD-9 disease symptoms [25], including cough, headache, chest pain) in each ED node and one potential anomalous connected subspace cluster (highlighted in the shaded region and blue-colored texts) that has a anomalous subset of attributes (cough and headache) and a connected subgraph of nodes (1, 2, 3, 5, and 6). The counts of these two attributes within the subgraph are abnormally higher than those outside the subgraph. The anomalous connected subspace cluster is used for disease outbreak detection.

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.13

TABLE I: Comparison of related work ("Generality" refers to the capability of a method to support different score functions and topological constraints on subspace clusters on attributed networks. "Good tradeoff" refers to the good trade-off between tractability and quality guarantee on subspace clusters, when the number of feasible subgraphs (neighborhoods) is large.

	Cluster detection	On attributed networks	Attribute subspace	Coherent dense clusters	Anomalous connected clusters	Good trade-off	Generality
METIS [21], Spectral [26], Co-clustering [11]	✓						
PICS [2], CODA [14]	✓	✓					
NPHGS [7], EDAN [35], CSGN [31], GSSO [39], GSPA [8]	✓	✓			✓		
CoPaM [24], Gamer [15], [16], `FocusCO` [30], AW-NCut [17]	✓	✓	✓	✓			
SODA [18], AMEN [29]	✓	✓	✓		✓		
`SG-Pursuit` [this paper]	✓	✓	✓	✓	✓	✓	✓

either utilize all the given attributes [14], [27] or perform a unsupervised feature selection as a preprocessing step [37]. However, as demonstrated in a number of studies [15]–[17], [29], [30], clusters of interest in a multi-attributed network are often **subspace clusters**, each of which is defined by **a cluster of nodes** and **a relevant subset of attributes**. For example, in social networks, it is very unlikely that people are similar within all of their characteristics [16]. In health surveillance networks, it is very rare that outbreaks of different disease types have identical symptoms [25]. In order to detect subspace clusters, it is required to conduct feature selection and cluster detection, concurrently, as without knowing the true clusters of nodes, it is difficult to identify their relevant attributes, and vice versa.

In recent years, a limited number of methods have been proposed to detect *subspace clusters*, which fall into two main categories, including detection of coherent dense subspace clusters and detection of anomalous connected subspace clusters. The methods for detecting **coherent dense subspace clusters** search for subsets of nodes that show high similarity in subsets of their attributes and that are as well densely connected within the input network. Customized algorithms are developed for specific combinations of similarity functions of attributes (e.g., threshold based [16], [17] and pairwise distance based [30] functions) and density functions of nodes [15]–[17], [24], [30]. The methods for detecting **anomalous connected subspace clusters** search for subsets of nodes that are significantly different from the other nodes on subsets of their attributes and that are as well connected (but not necessary dense) within the input network. The connectivity constraint ensures that the clusters of nodes reflect changes due to localized in-network processes. All the existing methods in this category consider a small set of neighborhoods (e.g., social circles and ego networks [29], subgraphs isomorphic to a query graph [18], and small-diameter subgraphs [25]), and identify anomalous subspace clusters *among only these given neighborhoods*.

However, the aforementioned methods have two main limitations: 1) **Lack of generality.** All these methods are customized for specific score functions of attributes and topological constraints on clusters, and may be inapplicable if the functions or constraints are changed. As discussed in recent surveys [1], the definition of an interesting subgraph pattern, in which subspace clusters is a specific type, is meaningful only under a given context or application. There is a strong need of generic methods that can handle a broad class of score functions, such as parametric/nonparametric scan statistic functions [7], discriminative functions [33], and least square functions [9]; and topological constraints, such as the types of subgraphs aforementioned [16], [18], [25], [29], [30], compact subgraphs [35], trees [22], and paths [3]. 2) **Lack of good tradeoff between tractability and quality guarantees.** The methods for detecting anomalous connected subgraphs conduct exhaust search over all feasible subgraphs (neighborhoods), but will be intractable when the number of feasible subgraphs is large (e.g., all connected subgraphs). Several methods for detecting coherent dense subspace clusters are tractable to large networks, but do not provide worst-case theoretical guarantees on the quality of the detected clusters.

This paper presents a novel generic and theoretical framework to address the above two main limitations of existing methods for a broad class of interesting subspace cluster detection problems. In particular, we consider the general form of subspace cluster detection as an optimization problem that has a general score function measuring the interestingness of a subset of features and a cluster of nodes, a sparsity constraint on the subset of features, and topological constraints on the cluster of nodes. We propose a novel subspace graph-structured matching pursuit algorithm, namely, `SG-Pursuit`, to approximately solve this general problem in nearly-linear time. The key idea is to iteratively search for a close-to-optimal solution by solving easier subproblems in each iteration, including i) identification of topological-free clusters of nodes and a sparsity-free subset of attributes that maximizes the score function in a sub-solution-space determined by the gradient of the current solution; and ii) projection of the identified intermediate solution onto the solution-space defined by the sparsity and topological constraints. The contributions of this work are summarized as follows:

- **Design of a generic and efficient approximation algorithm for the subspace cluster detection problem.** We propose a novel generic algorithm, namely, `SG-Pursuit`, to approximately solve a broad class of subspace cluster detection problems that are defined by different score functions and topological constraints in nearly-linear time. To the best of our knowledge, this is the first-known generic algorithm for such problems.

- **Theoretical guarantees and connections.** We present a theoretical analysis of the proposed `SG-Pursuit` and show that `SG-Pursuit` enjoys a geometric rate of convergence and a tight error bound on the quality of the detected subspace clusters. We further demonstrate that `SG-Pursuit` enjoys strong guarantees analogous to state-of-the-art methods for sparse feature selection in high-dimensional data and for subgraph detection in attributed networks.
- **Comprehensive experiments to validate the effectiveness and efficiency of the proposed techniques.** `SG-Pursuit` was specialized to conduct the specific tasks of coherent dense subspace cluster detection and anomalous connected subspace cluster detection on several real-world data sets. The results demonstrate that `SG-Pursuit` outperforms state-of-the-art methods that are designed specifically for these tasks, even though `SG-Pursuit` is designed to address general subspace cluster detection problems.

The rest of this paper is organized as follows. Section II introduces the proposed method `SG-Pursuit` and analyzes its theoretical properties. Section III discusses applications of our proposed algorithm for the tasks of coherent dense subspace cluster detection and anomalous connected subspace cluster detection. Experiments on several real-world benchmark datasets are presented in Section IV. Section V concludes the paper and describes future work.

II. METHOD SG-PURSUIT

In this section we first formulate the problem of subspace cluster detection formally. Next, we present the algorithm `SG-Pursuit` and analyze its theoretical properties, including its convergence rate, error-bound, and time complexity.

A. Problem Formulation

We consider a multi-attributed network that is defined as $\mathbb{G} = (\mathbb{V}, \mathbb{E}, w)$, where $\mathbb{V} = \{1, \cdots, n\}$ is the ground set of nodes of size n, $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is the set of edges, and the function $w : \mathbb{V} \to \mathbb{R}^p$ defines a vector of attributes of size p for each node $v \in \mathbb{V}$: $w(v) \in \mathbb{R}^p$. For simplicity, we denote the attribute vector $w(v)$ by w_v.

We introduce two vectors of coefficients, including $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^p$, that will be optimized for detecting the most interesting subspace cluster in $\mathbb{G}$, where x identifies the cluster (subset) of nodes and y identifies their relevant attributes. In particular, the vector x refers to the vector of coefficients of the nodes in $\mathbb{V}$. Each node $i \in \mathbb{V}$ has a coefficient score x_i indicating the importance of this node in the cluster of interest. If $x_i \neq 0$, it means that the node i belongs to the cluster of interest. Similarly, the vector y refers to the vector of coefficients of the p attributes. Each attribute $j \in \{1, \cdots, p\}$ has a coefficient score y_j indicating the relevance of this attribute to the clusters of interest. Let $\text{supp}(x)$ be the support set of indices of nonzero entries in x: $\text{supp}(x) = \{i \mid x_i \neq 0\}$. Then the support set $\text{supp}(x)$ represents the subset of nodes

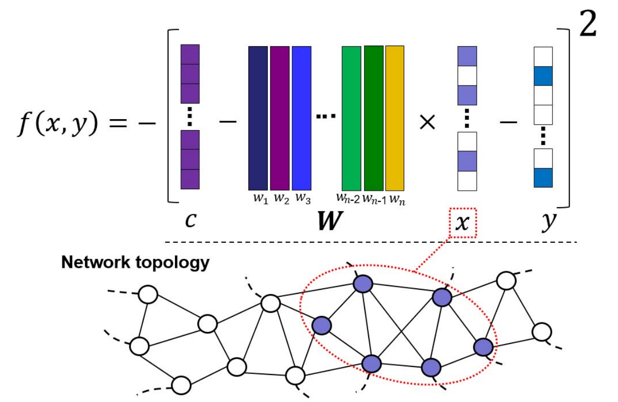

Fig. 3: An example function of $f(x, y)$ (negative squared error function (2)) for robust linear regression models that has been widely used in anomaly detection tasks [9], [34], [36], [38]. In this example, the vector x is a vector of sparse coefficients of the nodes in the input network that must satisfy the topological constraints ($\mathbb{M}(k = 6)$): the size of $\text{supp}(x)$ is at most 6. The residual vector y is sparse by the constraint $\|y\|_0 \leq s$ and is used to identify anomalous attributes.

that belong to the cluster of interest. The support set $\text{supp}(y)$ represents the subset of relevant attributes. We define the feasible space of clusters of nodes as

$$\mathbb{M}(k) = \{S \mid S \subseteq \mathbb{V}; |S| \leq k;\ \mathbb{G}_S \text{ satisfies predefined topological constraints.} \},$$

where S refers to a subset of nodes in $\mathbb{V}$, $\mathbb{G}_S = (S, \mathbb{E} \cap S \times S)$ refers to the subgraph induced by S, $|S|$ refers to the total number of nodes in S, and k refers to an upper bound on the size of the cluster. The topological constraints can be any topological constraints on $\mathbb{G}_S$, such as connected subgraphs [25], [29], dense subgraphs [16], [30], subgraphs that are isomorphic to a query graph [18], compact subgraphs [35], trees [22], and paths [3], among others.

Based on the above notations, we consider **a general form of the subspace cluster detection problem** as

$$\max_{x \in \mathcal{C}_x, y \in \mathcal{C}_y} f(x, y) \quad s.t. \quad \text{supp}(x) \in \mathbb{M}(k) \text{ and } \|y\|_0 \leq s, \quad (1)$$

where $f(x, y) : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ is a score function that measures the overall level of interestingness of the subspace clusters indicated by x and y; $\mathcal{C}_x \subseteq \mathbb{R}^n$ represents a convex set in the Euclidean space $\mathbb{R}^n$, $\mathcal{C}_y \subseteq \mathbb{R}^p$ represents a convex set in the Euclidean space $\mathbb{R}^p$, $\mathbb{M}(k)$ refers to the feasible space of clusters of nodes as defined a above, and s refers to an upper bound on the number of attributes relevant to the subspace clusters of interest. The parameters k and s are predefined by the user. Let $\hat{x}$ and $\hat{y}$ be the solution to Problem (1). Denote by S the support set $\text{supp}(\hat{x})$ that represents the most interesting cluster of nodes, and by R the support set $\text{supp}(\hat{y})$ represents the subset of relevant attributes. The most interesting subspace cluster can then be identified as (S, R).

As illustrated in Figure 3, an example score function $f(x, y)$ is a negative squared error function for robust linear regression that has been widely used in anomaly detection tasks [9], [34], [36], [38]:

$$f(x, y) = -\|c - W^T x - y\|_2^2, \quad (2)$$

where $x \in C_x := \mathbb{R}^n$, $y \in C_y := \mathbb{R}^p$, $c \in \mathbb{R}^p$ refers to a vector of observed response values, and $W = [w_1, w_2, \cdots, w_n]^T \in \mathbb{R}^{n \times p}$. The residual vector y is used to identify anomalous attributes, and its sparsity s is usually much smaller than p (the total number of attributes). There are also applications where both x and y need to be vectors of positive coefficients [38]: $C_x := \mathbb{R}_+^n$ and $C_y := \mathbb{R}_+^p$.

Remark 1. *There are scenarios where x is considered as a vector of binary values, instead of numerical coefficients, and the resulting problem becomes a discrete optimization problem that is NP-hard in general and does not have known solutions. In this case, by relaxing the input domain of x from $\{0,1\}^n$ to the convex set $\mathcal{C}_x := [0,1]^n$ and replacing the score function $f(x, y)$ with its tight concave surrogate function, the resulting relaxed problem becomes a special case of Problem (1). In particular, when the cost function is a supermodular function of x, a tight concave surrogate function can be obtained based on Lobasz extensions, such that the solutions to the relaxed problem are identical to the solutions to the original discrete optimization problem. In addition, the same equivalence also holds for a number of popular non-convex functions that are non-supermodular, such as Hinge and Squared Hinge functions, and their tight concave surrogate functions have been studied in recent work [6], [36].*

Remark 2. *Problem (1) considers the detection of the most interesting subspace cluster in a multi-attributed network. There are applications, where top g most interesting subspace clusters are of interest, where g is predefined by the user. In this case, the g clusters can be identified one-by-one, repeatedly, by solving Problem (1) for each subspace cluster, excluding the clusters that have been found.*

B. Head and Tail Projections on $\mathbb{M}(k)$

Before we present our proposed algorithm `SG-Pursuit`, we first introduce two major components related to the support of the topological constraints "supp$(x) \in \mathbb{M}(k)$", including head and tail projections. The key idea is that, suppose we are able to find a good intermediate solution $\hat{x}$ that does not satisfy this constraint, these two types of projections will be used to find good approximations of $\hat{x}$ in the feasible space defined by $\mathbb{M}(k)$.

- **Tail Projection** $\mathrm{T}(x)$ [20]: Find a $S \subseteq \mathbb{V}$ such that

$$\|x - x_S\|_2 \leq c_T \cdot \min_{S' \in \mathbb{M}(k)} \|x - x_{S'}\|_2, \quad (3)$$

  where $c_T \geq 1$, and x_S is the restriction of x to indices in S: we have $(x_S)_i = x_i$ for $i \in S$ and $(x_S)_i = 0$ otherwise. When $c_T = 1$, $\mathrm{T}(x)$ returns an optimal solution to the problem: $\min_{S' \in \mathbb{M}(k)} \|x - x_{S'}\|_2$. When $c_H < 1$, $\mathrm{T}(x)$ returns an approximate solution to this problem with the approximation factor c_T.

- **Head projection** $\mathrm{H}(x)$ [20]: Find a $S \subseteq \mathbb{V}$ such that

$$\|x_S\|_2 \geq c_H \cdot \max_{S' \in \mathbb{M}(k)} \|x_{S'}\|_2, \quad (4)$$

  where $c_H \in [0,1]$. When $c_H = 1$, $\mathrm{H}(x)$ returns an optimal solution to the problem: $\max_{S' \in \mathbb{M}(k)} \|x_{S'}\|_2$. When $c_H < 1$, $\mathrm{H}(x)$ returns an approximate solution to this problem with the approximation factor c_H.

It can be readily proved that, when $c_T = 1$ and $c_H = 1$, both $\mathrm{T}(x)$ and $\mathrm{H}(x)$ return the same subset S, and the corresponding vector x_S is optimum for the standard projection oracle in the traditional projected gradient descent algorithm [4]:

$$\arg\min_{x' \in \mathbb{R}^n} \|x - x'\|_2 \quad s.t. \quad \mathrm{supp}(x') \in \mathbb{M}(k), \quad (5)$$

which is NP-hard in general for popular topological constraints, such as connected subgraphs and dense subgraphs [31]. However, when $c_T > 1$ and $c_H < 1$, $\mathrm{T}(x)$ and $\mathrm{H}(x)$ return different approximate solutions to the standard projection problem (5). Although the head and tail projections are NP-hard problems when $c_T = 1$ and $c_H = 1$, these two projections can often be implemented in nearly-linear time when we allow relaxations on c_T and c_H: $c_T > 1$ and $c_H < 1$. For example, when the topological constraint considered in $\mathbb{M}(k)$ is that: "$\mathbb{G}_{S_i}$ is a connected subgraph", where S_i is a specific cluster of nodes, the resulting head and tail projections can be implemented in nearly-linear time with the parameters: $c_T = \sqrt{7}$ and $c_H = \sqrt{1/14}$ [20]. The impact of these two parameters on the performance of `SG-Pursuit` will be discussed in Section II-D.

As discussed above, the head and tail projections can be considered as two different approximations to the standard projection problem (5). It has been demonstrated that the joint utilization of both head and tail projections is critical in design of approximate algorithms for network-related optimization problems [8], [19], [20], [39].

C. Algorithm Details

We propose a novel `S`ubspace `G`raph-structured matching `Pursuit` algorithm, namely, `SG-Pursuit`, to approximately solve Problem (1) in nearly-linear time. The key idea is to iteratively search for a close-to-optimal solution by solving easier subproblems in each iteration i, including i) <u>identification</u> of the intermediate solution (b_x^i, b_y^i) that maximizes the score function $f(x, y)$ in a solution-subspace determined by the partial derivatives of the function on the current solution, including $\nabla_x f(x^i, y^i)$ and $\nabla_y f(x^i, y^i)$, and ii) <u>projection</u> of the intermediate solution (b_x^i, b_y^i) to the feasible space defined by the topological constraints: "supp$(x) \in \mathbb{M}(k)$", and the sparsity constraint: "$\|y\|_0 \leq s$". The projected solution (x^{i+1}, y^{i+1}) is then the updated intermediate solution returned by this iteration.

The main steps of `SG-Pursuit` are shown in Algorithm 1. The procedure generates a sequence of intermediate solutions (x^0, y^0), $(x^1, y^1), \cdots$, from an initial solution (x^0, y^0). At the i-th iteration, the first step (**Line 6**) calculates the partial derivative $\nabla_x f(x^i, y^i)$, and then identifies a subset of nodes via head projection that returns a support set with the head value at least a constraint factor c_H of the optimal head value: "$\Gamma_x = \mathrm{H}(\nabla_x f(x^i, y^i))$". The support set Γ_x can be interpreted as the directions where the nonconvex set "$\mathrm{supp}(x) \in \mathbb{M}(k)$" is located, within which pursuing the maximization over y will be most effective. The second step (**Line 7**) identifies the $2s$ nodes of the partial derivative vector $\nabla_y f(x^i, y^i)$ that have the largest magnitude that are chosen as the directions in which pursuing the maximization on y will be most effective: "$\Gamma_y = \arg\max_{R\subseteq\{1,\cdots,p\}}\{\|[\nabla_y f(x^i, y^i)]_R\|_2^2 : \|R\|_0 \le 2s\}$." The subsets Γ_x and Γ_y are then merged in **Line 8** and **Line 9** with the supports of the current estimates "$\mathrm{supp}(x^i)$" and "$\mathrm{supp}(y^i)$", respectively, to obtain "$\Omega_x = \Gamma_x \cup \mathrm{supp}(x^i)$" and "$\Omega_y = \Gamma_y \cup \mathrm{supp}(y^i)$". The combined support sets define a subspace of x and y over which the function $f(x, y)$ is maximized to produce an intermediate solution in **Line 10**: "$(b_x^i, b_y^i) = \arg\max_{x\in\mathcal{C}_x, y\in\mathcal{C}_y} f(x, y) \quad s.t. \quad \mathrm{supp}(x) \subseteq \Omega_x, \mathrm{supp}(y) \subseteq \Omega_y$." Then a subset of nodes is identified via tail projection of b_x^i in **Line 11**: "$\Psi_x^{i+1} = \mathrm{T}(b_x^i)$", that returns a support set with the tail value at most a constant c_T times larger than the optimal tail value. A subset of attributes of size s that have the largest magnitude is chosen in **Line 12** as the subset of relevant attributes: "$\Psi_y^{i+1} = \arg\max_{R\subseteq\{1,\cdots,p\}}\{\|[b_y^i]_R\|_2^2 : \|R\|_0 \le s\}$." As the final steps of this iteration (**Line 13** and **Line 14**), the estimates x^{i+1} and y^{i+1} are updated as the restrictions of b_x^i and b_y^i on the support sets Ψ_x^{i+1} and Ψ_y^{i+1}, respectively: "$x^{i+1} = [b_x^i]_{\Psi_x^{i+1}}$" and "$y^{i+1} = [b_y^i]_{\Psi_y^{i+1}}$." These steps are conducted to ensure that the estimates x^{i+1} and y^{i+1} returned by each iteration always satisfy the sparsity and topological constraints, respectively. After the termination of the iterations, **Line 17** identifies the subspace cluster: $\mathcal{C} = (\Psi_x^i, \Psi_y^i)$, where Ψ_x^i represents the subset (cluster) of nodes and Ψ_y^i represents the subset of relevant attributes.

Algorithm 1 `SG-Pursuit`

1: **Input**: Network instance $\mathbb{G}$ and the parameters, including k (maximum number of nodes in the subspace cluster) and s (maximum size of selected features).
2: **Output**: The vectors of coefficients of nodes and attributes, including x^i and y^i, and the identified subspace cluster $\mathcal{C}$.
3: $\epsilon = 0.0001$ % The termination criterion of iterations
4: $i = 0$; $x^i, y^i =$ initial vectors
5: **repeat**
6: $\quad \Gamma_x = \mathrm{H}(\nabla_x f(x^i, y^i))$
7: $\quad \Gamma_y = \arg\max_{R\subseteq\{1,\cdots,p\}}\{\|[\nabla_y f(x^i, y^i)]_R\|_2^2 : \|R\|_0 \le 2s\}$
8: $\quad \Omega_x = \Gamma_x \cup \mathrm{supp}(x^i)$
9: $\quad \Omega_y = \Gamma_y \cup \mathrm{supp}(y^i)$
10: $\quad (b_x^i, b_y^i) = \arg\max_{x\in\mathcal{C}_x, y\in\mathcal{C}_y} f(x, y) \quad \text{s.t.} \quad \mathrm{supp}(x) \subseteq \Omega_x,\ \mathrm{supp}(y) \subseteq \Omega_y$
11: $\quad \Psi_x^{i+1} = \mathrm{T}(b_x^i)$
12: $\quad \Psi_y^{i+1} = \arg\max_{R\subseteq\{1,\cdots,p\}}\{\|[b_y^i]_R\|_2^2 : \|R\|_0 \le s\}$
13: $\quad x^{i+1} = [b_x^i]_{\Psi_x^{i+1}}$
14: $\quad y^{i+1} = [b_y^i]_{\Psi_y^{i+1}}$
15: $\quad i = i + 1$
16: **until** $\|x^i - x^{i-1}\| \le \epsilon$ and $\|y^i - y^{i-1}\| \le \epsilon$
17: $\mathcal{C} = (\Psi_x^i, \Psi_y^i)$.
18: **return** $x^i, y^i, \mathcal{C}$

D. Theoretical Analysis

In order to demonstrate the accuracy and efficiency of `SG-Pursuit`, we require that the score function $f(x, y)$ satisfies the Restricted Strong Concavity/Smoothness (RSC/RSS) condition as follows:

Definition II.1 (Restricted Strong Concavity/Smoothness (RSC/RSS))**.** A score function f satisfies the $(\mathbb{M}(k), s, \gamma^-, \gamma^+)$-$RSS/RSC$ if, for every $x, x' \in \mathbb{R}^n$ and $y, y' \in \mathbb{R}^p$ with $\mathrm{supp}(x) \subseteq \mathbb{M}(2k)$, $\mathrm{supp}(x') \subseteq \mathbb{M}(2k)$, $|\mathrm{supp}(y)| \le 2s$, and $|\mathrm{supp}(y')| \le 2s$, the following inequalities hold:

$$\frac{\gamma^-}{2}\left(\|x - x'\|_2^2 + \|y - y'\|_2^2\right) \le$$
$$f(x', y') - f(x, y) - \nabla_x f(x, y)^T(x' - x) - \nabla_y f(x, y)^T(y' - y) \le$$
$$\frac{\gamma^+}{2}\left(\|x - x'\|_2^2 + \|y - y'\|_2^2\right).$$

The RSC/RSC condition basically characterizes cost functions that have quadratic bounds on the derivative of the objective function when restricted to the graph-structured vector x and the sparsity-constrained vector y. When the score function f is a quadratic function of x and y, RSC/RSC condition degeneralizes to the restricted isometry property (RIP) that is well-known in the field of compressive sensing. For example, we consider the negative squared error function (2) as discussed in Section II-A: $f(x, y) = -\|c - W^T x - y\|_2^2$. Let $\bar{W} = [W^T, I]$, where I is a p by p identity matrix. Let $z = [x^T, y^T]^T$. The RSC/RSC condition can be reformulated as the RIP condition:

$$(1 - \delta)\|z\|_2^2 \le \|\bar{W} z\|_2^2 \le (1 + \delta)\|z\|_2^2,$$

where $\gamma^+ = 2(1 + \delta)$, $\gamma^- = 2(1 - \delta)$, and $\delta \in (0, 1)$ is the standard parameter as defined in RIP. However, the RIP condition in this example is different from the traditional RIP condition in that the components of z, including x and y, must satisfy the constraints related to $\mathbb{M}(k)$ and the sparsity s as described in Definition II.1.

Theorem II.1. *If the score function f satisfies the property $(\mathbb{M}(k),\ s, \gamma^-, \gamma^+)$-$RSS/RSC$, then for any true $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^p$, the iterations of the proposed algorithm `SG-Pursuit` satisfy the inequality*

$$\|r_x^{i+1}\|_2 + \|r_y^{i+1}\|_2 \le \alpha\Big(\|r_x^i\|_2 + \|r_y^i\|_2\Big) + \beta(\varepsilon_x + \varepsilon_y)$$

where $r_x^{i+1} = x^{i+1} - x^*$, $r_y^{i+1} = y^{i+1} - y^*$, $\alpha_0 = c_H(1 - \rho) - \rho$, $\rho = \sqrt{1 - (\frac{\gamma^-}{\gamma^+})}$, $\beta_0 = (c_H + 1)\frac{\gamma^-}{(\gamma^+)^2}$, $\alpha = \frac{(c_T+1)\sqrt{2-2\alpha_0^2}}{1-\sqrt{2}\rho}$, $\beta = \frac{(c_T+1)}{1-\sqrt{2}\rho}\Big(\frac{\gamma^-}{(\gamma^+)^2} + \Big(\frac{\sqrt{2}\alpha_0\beta_0}{1-\alpha_0^2} + \frac{\sqrt{2}\beta_0}{\alpha_0}\Big)\Big)$, $\varepsilon_x = \max_{S\in\mathbb{M}(2k)}\|[\nabla f_x(x^*, y^*)]_S\|_2^2$, *and* $\varepsilon_y = \max_{R\subseteq\{1,\cdots,p\},|R|\le 3s}\|[\nabla f_y(x^*, y^*)]_R\|_2^2$.

Proof: See the technical report [12] for details. ■

Theorem II.2. *Let (x^*, y^*) the optimal solution to Problem (1) and f be a score function that satisfies the $(\mathbb{M}(k), s, \gamma^-, \gamma^+)$-RSS/RSC property. Let T and H be the tail and head projections with c_T and c_H such that $0 < \alpha < 1$. Then after $t = \left\lceil \log \left(\frac{\|x^*\|_2 + \|y^*\|_2}{\varepsilon_x + \varepsilon_y} \right) / \log \frac{1}{\alpha} \right\rceil$ iterations,* `SG-Pursuit` *returns a single estimate $(\hat{x}, \hat{y})$ satisfying*

$$\|\hat{x} - x^*\|_2 + \|\hat{y} - y^*\|_2 \leq c(\varepsilon_x + \varepsilon_y), \tag{6}$$

where $c = (1 + \frac{\beta}{1-\alpha})$ is a fixed constant. Moreover, `SG-Pursuit` *runs in time*

$$O\left((T_1 + T_2 + p \log p) \log \left((\|x^*\|_2 + \|y^*\|_2)/(\varepsilon_x + \varepsilon_y) \right) \right), \tag{7}$$

where T_1 is the time complexity of one execution of the subproblem in Line 10 in `SG-Pursuit` *and T_2 is the time complexity of one execution of the head and tail projections.*

In particular, when the connectivity constraint or a density constraint is considered as the topological constraint on the feasible clusters of nodes in $\mathbb{M}(k)$, there exist efficient algorithms for the head and tail projections that have the time complexity $O(|\mathbb{E}| \log^3 n)$ [12], [20]. When s and k are fixed small constants with respect to n, the subproblem in Line 10 in `SG-Pursuit` *can be solved in nearly linear time in practice using convex optimization algorithms, such as the project gradient descent algorithm. Therefore, under these conditions, for coherent dense subgraph detection and connected anomalous subspace cluster detection problems,* `SG-Pursuit` *has a nearly-linear time complexity on the network size n and the cardinality of attributes p:*

$$O\left((|\mathbb{E}| \log^3 n + p \log p) \log \left((\|x^*\|_2 + \|y^*\|_2)/(\varepsilon_x + \varepsilon_y) \right) \right). \tag{8}$$

Proof: From Theorem II.1, the following inequality can be obtained via an inductive argument: $\|x^i - x^*\|_2 + \|y^i - y^*\|_2 \leq \alpha^i(\|x^*\|_2 + \|y^*\|_2) + \beta(\varepsilon_x + \varepsilon_y) \sum_{j=0}^{i} \alpha^i$. For $i = \left\lceil \log \left(\frac{\|x^*\|_2 + \|y^*\|_2}{\varepsilon_x + \varepsilon_y} \right) / \log \frac{1}{\alpha} \right\rceil$, we have $\alpha^i(\|x^*\|_2 + \|y^*\|_2) \leq (\varepsilon_x + \varepsilon_y)$. The geometric series $\sum_{j=0}^{i} \alpha^i$ can be bounded by $\frac{1}{1-\alpha}$. The error bound (6) can be obtained by combining the preceding inequalities. The time complexity of the subproblem in Line 10 is denoted by $O(T_1)$, and the time complexities of both head and tail projections are denoted by $O(T_2)$. The time complexity to solve the subproblem in Line 7 is $O(p \log p)$, as the exact solution can be obtained by sorting the entries in $\nabla_y f(x^i, y^i)$ in a descending order based their absolute values, and then returning the indices of the top $2s$ entries. Similarly, the time complexity to solve the subproblem in Line 12 is $O(p \log p)$. As the total number of iterations is $O \log(\frac{\|x^*\|_2 + \|y^*\|_2}{\varepsilon_x + \varepsilon_y})$, the time complexity specified in Equation (7) can be calculated. accordingly. When T_1 is bounded by $O(n \log n)$ and $T_2 = |\mathbb{E}| \log^3 n$, the nearly-linear time complexity specified in Equation (8) can be obtained. ■

Theorem II.2 shows that `SG-Pursuit` enjoys a geometric rate of convergence and the estimation error is determined by the multiplier of $(\varepsilon_x + \varepsilon_y)$, where $\varepsilon_x = \max_{S \in \mathbb{M}(8k)} \|[\nabla f_x(x^*, y^*)]_S\|_2^2$, and $\varepsilon_y = \max_{R \subseteq \{1, \cdots, p\}, |R| \leq 2s} \|[\nabla f_y(x^*, y^*)]_R\|_2^2$. The shrinkage rate $\alpha < 1$ controls the converge rate of `SG-Pursuit`. In particular, if the true x^* and y^* are sufficiently close to an unconstrained maximum of f, then the estimation error is negligible because both ε_x and ε_y have small magnitudes. Especially, in the ideal case where $\varepsilon_x = \varepsilon_y = 0$, it is guaranteed that we can obtain the true x and y to arbitrary precision. Note that we can make γ^+ and γ^- as close as we desire, such that $\gamma^+/\gamma^- \approx 1$, since this assumption only affects the measurement bound by a constraint factor. In this case, in order to ensure that $\alpha \approx (c_T + 1)\sqrt{2 - 2c_H^t} < 1$, the factors c_T and c_H should satisfy the inequality: $c_H^2 > 1 - 1/\left(2(1 + c_T)^2\right)$. As proved in [19], the fixed factor c_H of any given algorithm for the head projection can be boosted to any arbitrary constant $c'_H < 1$, such that the above condition can be satisfied. This indicates the flexibility of designing approximate algorithms for head and tail projections in order to ensure the geometric convergence rate of `SG-Pursuit`.

Remark 3. *(Connections to existing methods)* `SG-Pursuit` *is a generalization of the* `GraSP` *(Gradient Support Pursuit) method [5], a state-of-the-art method for general sparsity-constrained optimization problems, and the* `Graph-MP` *method [8], a state-of-the-art method for general graph-structured sparse optimization problems. In particular, when we fix x and only update y in the steps of* `SG-Pursuit`*,* `SG-Pursuit` *degeneralizes to* `GraSP`*. When we fix y and only update x in the steps of* `SG-Pursuit`*,* `SG-Pursuit` *degeneralizes to* `Graph-MP`*. Surprisingly, even that* `SG-Pursuit` *concurrently optimizes x and y, its convergence rate is of the same order as those of* `Graph-MP` *and* `GraSP` *under the RSS/RSC property [12].*

III. Example Applications

In this section, we specialize `SG-Pursuit` to address two typical subspace cluster detection problems in multi-attributed networks, including coherent dense subspace cluster detection and anomalous connected subspace cluster detection. The former searches for subsets of nodes that show high similarity in subsets of their attributes and that are as well densely connected within the input network. The coherence score function, as shown in Table II, is defined as the log likelihood ratio function, $\log \frac{\text{Prob}(\text{Data}|H_1(x,y))}{\text{Prob}(\text{Data}|H_0)}$, that corresponds to the hypothesis testing framework:

- Under the null (H_0), $w_{i,j} \sim \mathcal{N}(0, 1), \forall i \in \mathbb{V}, j \in \{1, \cdots, p\}$, where $w_{i,j}$ refers to the observed value of the j-th attribute of node i;
- Under the alternative $H_1(x, y)$, $w_{i,j} \sim \mathcal{N}(\mu_j, 1)$, if $x_i = 1$ and $y_j = 1$; otherwise, $w_{i,j} \sim \mathcal{N}(0, \sigma)$, where $x \in \{0, 1\}^n$, $y \in \{0, 1\}^p$, and $x_i = 1$ indicates that node i belongs to the cluster, $y_j = 1$ indicates that the attribute j belongs to the subset of coherent attributes. Each coherent attribute j has a different mean μ_j and the variance σ should be less than 1 (the variance of an incoherent attribute) in order to ensure the coherence of its observations. σ is set 0.01 by default.

The latter (anomalous connected subspace cluster detection) searches for subsets of nodes that are significantly different from the other nodes on subsets of their attributes and that are as well connected within the input network. The elevated mean scan statistic, as shown in Table II, is defined as the log likelihood ratio function that corresponds to a hypothesis testing framework that is the same as the above, except that 1) "coherent" is replaced by "anomalous", 2) the mean of each anomalous attribute is greater than (or more anomalous than) 0, the mean of a normal attribute, and the standard deviation of each anomalous attribute is set to 1 (the same variance of a normal attribute). The Fisher test statistic function is considered when each $w_{i,j}$ represents the level of anomalous (e.g., negative log p-value) of the j-th attribute of node i, and $x^T W y$ represents the overall level of anomalous of x and y. A large class of scan statistic functions for anomaly detection can be transformed to the fisher test statistic function using a 2-step procedure as proposed in [31]. The negative squared error is considered as the score function of anomalous subspace cluster detection in a regression setting and is introduced in Section II-A.

Theorem III.1. *When the attribute matrix W satisfies certain properties, the score functions, including the elevated mean scan statistic, the Fisher's test statistic, the negative square error, and the logistic function, satisfy the RSS/RSC property as described in Definition II.1.*

Proof: See the technical report [12] for details. ■

Theorem III.1 demonstrates that the theoretical guarantees of SG-Pursuit as analyzed in Section II-D are applicable to a number of popular score functions for subspace cluster detection problems. We note that SG-Pursuit also performs well in practice on the score functions not satisfying the RSS/RSC property as demonstrated in Section IV using the coherence score function as shown in Table II.

TABLE II: Example score (interestingness) functions.

Score Function[1]	Definition
Coherence score	$x^T\Big(W \odot W\Big)y - \frac{1}{0.01}x^T\Big(W - 1\frac{x^T W}{1^T x}\Big) \odot \Big(W - 1\frac{x^T W}{1^T x}\Big)y - \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2$
Elevated mean scan statistic	$x^T W y/\sqrt{x^T \mathbf{1}} - \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2$
Fisher's test statistic	$x^T W y - \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2$
Negative square error	$-\|c - W^T x - y\|_2^2 - \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2$
Logistic function	$\sum_{i=1}^{p}\Big(y_i \log g(x^T w_i) + (1 - y_i)\log(1 - g(x^T w_i))\Big) - \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2$

[1] The L2 regularization component "$-\frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2$" is considered in each score function to enforce the stability of maximizing the score function. $W = [w_1, \cdots, w_n]$, $\odot$ is a Hadamard product operator, and $g(z) = 1/(1+e^{-z})$.

IV. Experiments

This section evaluates the performance of our proposed method on the quality of the detected subspace clusters and run-time on synthetic and real-world networks. The experimental code and data sets are available from the Link [13].

A. Coherent dense subgraph detection

1) Experimental design: We compared SG-Pursuit with two representative methods, including GAMer [16] and FocusCO [30].

1. Generation of synthetic graphs: We used the same generator of synthetic coherent and dense subgraphs as used in the state-of-the-art FocusCO method [30], except that the standard deviation (std) of coherent attributes was set to $\sqrt{0.001}$, instead of 0.001, which makes the detection problem more challenging. The settings of the other parameters used in FocusCO include: $p_{in} = 0.35$ (density of edges in each cluster) and $p_{out} = 0.1$ (density of edges between clusters). We compared the performance of different methods based on different combinations of parameters: 1) the number of incoherent clusters, 2) the number of coherent attributes, 3) the total number of attributes, and 4) cluster size. We set these parameters to 9, 10, 100, 30, respectively, by default. We set the size of all coherent and incoherent clusters to 30, as GAMer was not scalable in the synthetic graphs for clusters of size larger than 30. We generated one coherent dense cluster and multiple incoherent dense clusters in each synthetic graph.

2. Real-world data. We used five public benchmark real-world attributed network datasets, including DBLP, Arxiv, Genes, IMDB, and DFB (German soccer premier league data), which are available from and described in details in [40]. The basic statistics of these five datasets are provided in Table III, with the numbers of nodes ranging from 100 to 11,989; the numbers of edges ranging from 1,106 to 119,258; and the number of attributes ranging from 5 to 300.

3. Implementation and parameter tuning: The implementations of FocusCO and Gamer are publicly released by authors [1]. **FocusCO** requires an exemplar set of nodes and has a trade-off parameter γ that is used in learning of feature weights. In order to make FocusCO the best competitive to our method, we used a random set of 90% nodes in each coherent dense subspace cluster as the input exemplar set of nodes. FocusCO estimates a weight for each attribute that characterizes the importance of this attribute, and return the top s attributes with the largest weights as the set of coherent attributes, and set s to the true number of coherent attributes. **GAMer** has four main parameters, including s_{min} (the minimum number of coherent attributes), γ_{min} (the minimum threshold on density), and n_{min} (the minimum cluster size), and w (the maximum width that control the level of coherence). We followed the recommended strategies by the authors and identified the best parameter values for FocusCO and Gamer. We did not consider other related methods that focus on different objectives rather than only cluster density (e.g., the normalized subspace graph cut objective as considered in [17]) and also their implemenations are not publically available.

4. Settings of our proposed method SG-Pursuit. We used the following score function to detect the most coherent dense subspace cluster in each synthetic graph: $f(x,y) = x^T\Big(W \odot W\Big)y - \frac{1}{0.01}x^T\Big(W - 1\frac{x^T W}{1^T x}\Big) \odot \Big(W - 1\frac{x^T W}{1^T x}\Big)y - \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2 + \lambda\frac{x^T A x}{1^T x}$, where A is the adjacency matrix of the input graph, and λ is a tradeoff parameter to balance to coherence score (See Table II) and the density score $\frac{x^T A x}{1^T x}$. The parameter λ was set to 5. We applied projected gradient

[1] Available at https://github.com/phanein/focused-clustering and http://dme.rwth-aachen.de/en/gamer

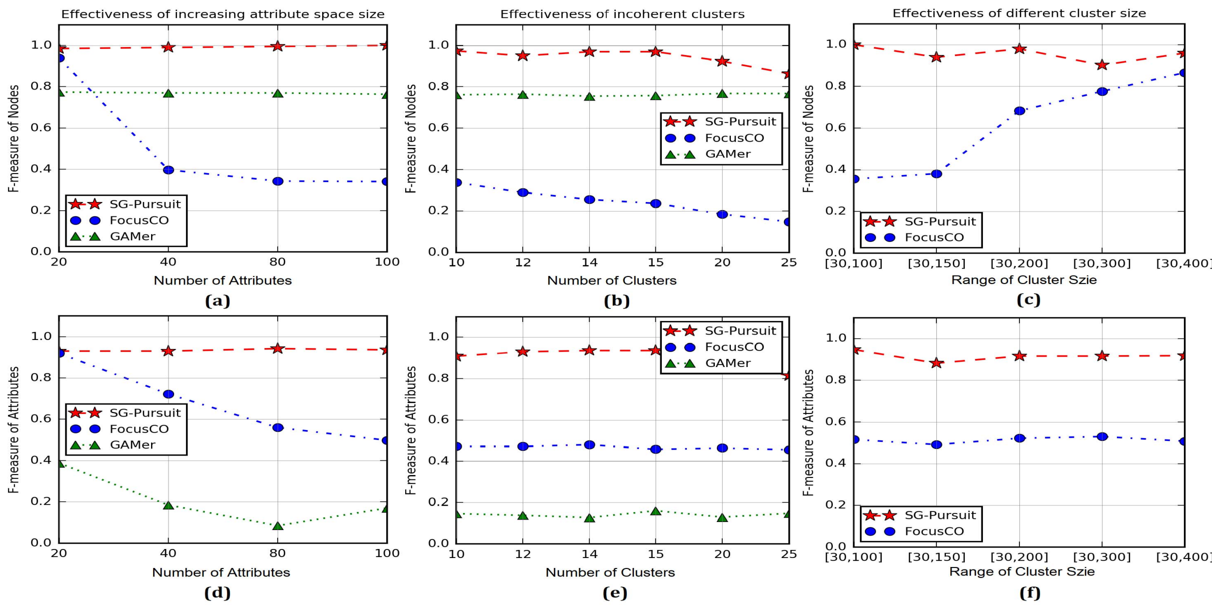

Fig. 4: Comparison on F-measures of detected nodes (first line) and of detected attributes (second line): (a) and (d) for changing the total number of attributes (with 10 coherent attributes); (b) and (e) for increasing number of clusters (with one coherent cluster); and (c) and (f) for changing cluster size variance (graph has variable size clusters), and measurements of `Gamer` are not shown in these two sub-figures as `Gamer` is unscalable to detection of clusters of size above 30.

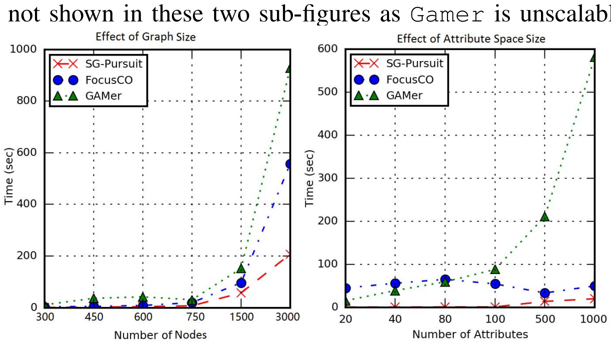

Fig. 5: Comparison on running time of synthetic graphs.

descent to solve the subprobelm in Line 10 of `SG-Pursuit`. The parameters k (upper bound of the cluster size) and s (upper bound on coherent attributes) were set to the true cluster size and number of coherent attributes, respectively, in the synthetic datasets. For the real-world datasets, we tested the ranges $k \in \{5, 10, 15, 30\}$ and $s \in \{3, 5, 10, 15, 20\}$ and identified the settings with the best objective scores. We note that `SG-Pursuit` does not require training data with labeled sub-space clusters to tune the parameters k and s.

5. Evaluation metrics. Each **synthetic graph** has a single true coherent dense subspace cluster (a combination of a subset of nodes and a subset of attributes) and the task was to detect this cluster. We reported the F-measures of the subsets of nodes and attributes for each competitive method. We note that `FocusCO` and `GAMer` may return multiple candidate clusters in an input graph, and in this case we return the cluster with the highest F-measure in order to make fair comparisons. We generated 50 synthetic graphs for each setting and reported the average F-measure and running time. For the five **real-world attributed network datasets**, where no ground truth is given, we considered three major measures, including **average cluster density**, **average cluster size**, and **average coherence distance**. The average cluster density is defined as the average degree of nodes within the K subspace clusters identified, where K is predefined. The coherence distance of a specific subspace cluster is defined as the average Euclidean distance between the nodes in this cluster based on the subset of attributed selected. The average coherence distance is the average of the coherence distances of the K subspace clusters. A combination of a high average cluster density, a high average cluster size, and a low average coherence distance indicates a high overall quality of the clusters detected.

2) Quality Analysis: 1) **Synthetic data with ground truth labels**. The comparison on F-measures among the three competitive methods is shown in Figure 4, by varying total number of irrelevant attributes, number of incoherent clusters, and cluster size variance. The results indicate that `SG-Pursuit` significantly outperformed `FocusCO` and `GAMer` with more than 15 percent marginal improvements in overall on F-measures of the detected nodes and the detect coherent attributes. As shown in Figure 4(c), when the cluster size increases, the F-measure of `FocusCO` consistently increases. In particular, we observed that when the cluster size is above 150, `FocusCO` achieved F-measure close to 1.0. In addition, when the standard deviation of coherent attributes decreases (in the shown Figures, we fixed this to $\sqrt{0.001}$), `FocusCO` performed better for large cluster sizes. To summarize, `SG-Pursuit` was more robust to `FocusCO` on low levels of coherence and small cluster sizes. More detailed comparison results are provided in our technical report [12]. 2) **Real-world data**. As the real-world datasets do not have ground truth labels, we can not apply `FocusCO` since it requires a predefined subset of ground truth nodes. Hence, we focus on the comparison between `SG-Pursuit` and `GAMer` with different predefined numbers of clusters (Top-$K, K = 5, 10, 15, 20$) . As shown in Table III, `SG-Pursuit` was able to identify subspace clusters with the three major measures coherently better than those of the clusters returned by `GAMer` in most of the settings. `GAMer` was able to identify clusters with densities larger than those detected by `SG-Pursuit`, but with much smaller cluster sizes and much large coherence distances.

3) Scalability analysis: The comparison on running times of competitive methods is shown in Figure 5 with respect to varying numbers of attributes and nodes of synthetic graphs. The results indicate that `SG-Pursuit` was faster than both `FocusCO` and `GAMer` over several orders of magnitude. The running time of `FocusCO` was independent of the number of attributes, but increases quadratically on the number of nodes (graph size). The running time `GAMer` increases quadratically on both numbers of attributes and nodes.

B. Anomalous connected cluster detection

1) Experimental design: We considered two representative methods, including `AMEN` [29] and `SODA` [18].

1. Data sets: 1) **Chicago Crime Data**. A data set of crime data records in Chicago was collected form the official website “https://data.cityofchicago.org/” from 2010 to 2014 that has 1,515,241 crime records in total, each of which has the location information (latitude and longitude), crime category (e.g., BATTERY, BURGLARY, THEFT), and description (e.g., “aggravated domestic battery: knife / cutting inst”). There are 35 different crime categories in total. We collected the census-tract-level graph in Chicago from the same website that has 46,357 nodes (census tracts) and 168,020 edges (there exists an edge if two census tracts neighbor with each other) in total, and considered the frequency of each keyword in the descriptions of crime records as an attribute. There are 121 keywords in total that are non-stop-words and have frequencies above 10,000, which are considered as attributes. In order to generate a ground-truth anomalous connected cluster of nodes, we picked a particular crime type (BATTERY or BURGLARY), identified a connected subgraph of size 100 via random walk, and then removed the crime records of this particular category in all nodes outside this subgraph, which generated a rare category as an anomalous category. This subgraph was considered as an anomalous cluster for crime records of categories that are different from this specific category, and the keywords that are specifically relevant to this category were considered as ground-truth anomalous attributes. We tried this process 50 times to generate 50 anomalous connected clusters, and manually identified 22 keywords relevant to BATTERY and 5 keywords relevant to BURGLARY as anomalous attributes. 2) **Yelp Data**. A Yelp reviews data set was publicly released by Yelp for academic research purposes[2]. All restaurants and reviews in the U.S. from 2014 to 2015 were considered, which includes 25,881 restaurants and 686,703 reviews. The frequencies of 1,151 keywords in the reviews that are non-stop-words and have frequencies above 5,000 are considered as attributes. We generated a geographic network of restaurants (nodes), in which each restaurant is connected to its 10 nearest restaurants, and there are 244,012 edges in total. We used the sample strategy as in the Chicago Crime Data to generate 50 ground-truth anomalous connected clusters of size 100 for the specific category “Mexican”. **2. Implementation and parameter tuning:** The implementations of `AMEN` and `SODA` are publicly released by the authors[3]. Their parameters were tuned by the recommended strategies by the authors. For our proposed method `SG-Pursuit`, we considered the elevated mean statistic function as defined in Table II. The upper bound of cluster size k was set to 100. The upper bound of number of attributes s was set to 22 for BATTERY related anomalous clusters and 5 for BURGLARY related anomalous clusters.

2) Quality and Scalability Analysis: The detection results of the competitive methods on the Chicago Crime Data are shown in Table IV. The results indicate that `SG-Pursuit` outperformed `SODA` and `AMEN` on F-Measure of nodes with more than 20% marginal improvements, and on F-measure of attributes with around 15% marginal improvements. The running time of `SG-Pursuit` was less than those of `SODA` and `AMEN` on several orders of magnitude. The results of our method on Yelp Data contain three parts: 1) The quality of returned clusters: The F-measure of the returned clusters is 0.31 with the precision 0.314 and the recall 0.309; 2) The top 10 most frequent keyword pairs, i.e. (frequency, keyword), returned are (21, “tacos”), (21, “asada”) (20, “taco”), (19, “salsa”) , (19, “level”), (15, “vegas”) (14, “mexican”), (14, “item”) (14, “beans”), and (13, “worth”), where the frequency of a keyword refers to the number of times that this keyword occurs in the anomalous subspace clusters detected by `SG-Pursuit`. 6 out of 10 keywords are related to "Mexican", which demonstrates that our method can identify the related keywords on the specified category; 3) The running time of our algorithm was 6.98 minutes. We were not able to obtain results from `AMEN` and `SODA` after running several hours. These baseline methods cannot handle graphs that have more than 10,000 nodes and 1,000 attributes.

V. CONCLUSIONS

This paper presents `SG-Pursuit`, a novel generic algorithm to subspace cluster detection in multi-attributed networks with both theoretical and empirical validations. Although `SG-Pursuit` enjoys rigorous guarantees when the score function satisfies RSC/RSS conditions, we observe that a number of score functions that have been used for subspace cluster detection, including the ones considered in the experiments, do not satisfy RSC/RSS conditions. For the future work, we plan to analyze theoretical properties of `SG-Pursuit` for score functions under conditions that are weaker than RSC/RSS con-

[2]Available at http://www.yelp.com/dataset_challenge

[3]Available at https://github.com/phanein/amen/tree/master/amen and https://github.com/manavs19/subgraph-outlier-detection

TABLE III: Analysis of five real-world datasets for coherent dense subspace cluster (subgraph) detection.

Dataset	Node	Edge	Attribute	Top-K	Avg. Cluster density		Avg. Cluster Size		Avg. Coherence Distance	
					SG-Pursuit	GAMer	SG-Pursuit	GAMer	SG-Pursuit	GAMer
DFB	100	1106	5	5	**9.69**	5.4	**10.8**	6.4	**0.13**	6.38
				10	**9.94**	3.9	**11.0**	4.9	**0.13**	7.41
				15	**11.03**	3.33	**12.07**	4.33	**0.12**	7.51
				20	**10.55**	2.9	**11.6**	3.9	**0.12**	8.25
DBLP	774	1757	20	5	**4.12**	3.2	**5.2**	4.2	**0.0**	**0.0**
				10	**3.84**	3.2	**5.7**	4.2	0.06	**0.0**
				15	**3.25**	3.17	**7.3**	4.17	0.3	**0.0**
				20	**3.34**	3.17	**7.68**	4.17	0.4	**0.0**
IMDB	862	4388	21	5	3.24	**4.96**	4.6	**10**	**0.0**	**0.0**
				10	3.17	**4.52**	5.67	**10.0**	0.08	**0.0**
				15	**3.52**	2.97	**5.79**	4.27	0.06	**0.0**
				20	**3.36**	2.78	**5.37**	4.2	0.06	**0.0**
Genes	2900	8264	115	5	3.76	**6.92**	**24.8**	10.6	**0.29**	8.28
				10	3.74	**5.91**	**24.7**	10.8	**0.31**	5.46
				15	3.82	**5.58**	**23.09**	10.87	**0.33**	4.69
				20	3.85	**5.36**	**21.83**	10.9	**0.33**	4.2
Arxiv	11989	119258	300	5	**11.53**	4.16	**15.8**	10.0	**0.0**	**0.0**
				10	**9.65**	4.24	**12.4**	10.0	**0.0**	**0.0**
				15	**9.03**	4.23	**11.27**	10.0	**0.0**	**0.0**
				20	**8.72**	4.21	**10.7**	10.0	**0.0**	**0.0**

TABLE IV: Chicago Crime data (Fm refers to F-Measure).

Methods	type	Node Fm	Attribute Fm	Running Time (s)
SODA	BATTERY	0.476	0.146	7,997.893
	BURGLARY	0.45	0.020	11,043.892
AMEN	BATTERY	0.363	0.818	3,835.589
	BURGLARY	0.337	0.800	4,265.449
SG-Pursuit	BATTERY	**0.683**	**0.955**	**73.998**
	BURGLARY	**0.538**	**1.000**	**37.538**

ditions. We also plan to generalize `SG-Pursuit` to subspace cluster detection in heterogeneous networks.

REFERENCES

[1] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *DMKD*, 29(3):626–688, 2015.

[2] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. Pics: Parameter-free identification of cohesive subgroups in large attributed graphs. In *SDM*, pages 439–450. SIAM, 2012.

[3] M. Asteris, A. Kyrillidis, A. Dimakis, H.-g. Yi, and B. Chandrasekaran. Stay on path: Pca along graph paths. In *ICML*, pages 1728–1736, 2015.

[4] S. Bahmani, P. T. Boufounos, and B. Raj. Learning model-based sparsity via projected gradient descent. *IEEE TIT*, 62(4):2092–2099, 2016.

[5] S. Bahmani, B. Raj, and P. T. Boufounos. Greedy sparsity-constrained optimization. *The JLMR*, 14(1):807–841, 2013.

[6] M. Ballerstein. *Convex relaxations for mixed-integer nonlinear programs*. Cuvillier, 2013.

[7] F. Chen and D. B. Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *KDD*, pages 1166–1175. ACM, 2014.

[8] F. Chen and B. Zhou. A generalized matching pursuit approach for graph-structured sparsity. In *Proc. IJCAI*, pages 1389–1395, 2016.

[9] X.-H. Dang, A. Silva, et al. Outlier detection from network data with subnetwork interpretation. In *ICDM*. IEEE, 2016.

[10] M. W. Davidson, D. A. Haim, and J. M. Radin. Using networks to combine “big data” and traditional surveillance to improve influenza predictions. *Scientific reports*, 5:8154, 2015.

[11] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD*, pages 89–98. ACM, 2003.

[12] Technical report. A Generic Framework for Interesting SubspaceCluster Detection in Multi-attributed Networks. Arxiv, 2017.

[13] Dataset and code. https://github.com/newera912/SG-Pursuit, 2017.

[14] J. Gao, F. Liang, et al. On community outliers and their efficient detection in information networks. In *KDD*, pages 813–822, 2010.

[15] S. Gunnemann, I. Farber, B. Boden, and T. Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *ICDM*, pages 845–850. IEEE, 2010.

[16] S. Günnemann, I. Färber, et al. Gamer: a synthesis of subspace clustering and dense subgraph mining. *KIS*, 40(2):243–278, 2014.

[17] S. Gunnemann, I. Farber, et al. Spectral subspace clustering for graphs with feature vectors. In *ICDM*, pages 231–240. IEEE, 2013.

[18] M. Gupta, A. Mallya, et al. Local learning for mining outlier subgraphs from network datasets. In *SDM*, pages 73–81. SIAM, 2014.

[19] C. Hegde, P. Indyk, and L. Schmidt. Approximation algorithms for model-based compressive sensing. *IEEE Transactions on Information Theory*, 61(9):5129–5147, 2015.

[20] C. Hegde, P. Indyk, and L. Schmidt. A nearly-linear time framework for graph-structured sparsity. In *ICML*, pages 928–937, 2015.

[21] G. Karypis and V. Kumar. Multilevel algorithms for multi-constraint graph partitioning. In *ACM Supercomputing*, pages 1–13,1998.

[22] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Convex and network flow optimization for structured sparsity. *JLMR*, 122681–2720, 2011.

[23] Z. Mihály, M. Kormos, et al. A meta-analysis of gene expression-based biomarkers predicting outcome after tamoxifen treatment in breast cancer. *Breast cancer research and treatment*, 140(2):219–232, 2013.

[24] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM*, pages 593–604,2009.

[25] D. B. Neill, E. McFowland, and H. Zheng. Fast subset scan for multivariate event detection. *Stat. in medicine*, 32(13):2185–2208, 2013.

[26] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, pages 849–856, 2001.

[27] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.

[28] F. Pasqualetti, F. Dörfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[29] B. Perozzi and L. Akoglu. Scalable anomaly ranking of attributed neighborhoods. In *SDM*, pages 207–215, 2016.

[30] B. Perozzi, L. Akoglu, et al. Focused clustering and outlier detection in large attributed graphs. In *KDD*, pages 1346–1355. ACM, 2014.

[31] J. Qian, V. Saligrama, and Y. Chen. Connected sub-graph detection. In *AISTATS*, volume 14, pages 22–25, 2014.

[32] N. Ramakrishnan, P. Butler, et al. ’beating the news’ with embers: forecasting civil unrest using open source indicators. In *KDD*, pages 1799–1808. ACM, 2014.

[33] S. Ranu, M. Hoang, and A. Singh. Mining discriminative subgraphs from global-state networks. In *KDD*, pages 509–517. ACM, 2013.

[34] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

[35] P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti. Event detection in activity networks. In *KDD*, pages 1176–1185. ACM, 2014.

[36] I. Shcherbatyi and B. Andres. Convexification of learning from constraints. In *GCPR*, pages 79–90. Springer, 2016.

[37] J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *KDD*, pages 904–912. ACM, 2012.

[38] H. Tong and C.-Y. Lin. Non-negative residual matrix factorization with application to graph anomaly detection. In *SDM*, pages 143–153, 2011.

[39] B. Zhou and F. Chen. Graph-structured sparse optimization for connected subgraph detection. In *ICDM*, 2016.

[40] Links to benchmark datasets used in this paper: http://dme.rwth-aachen.de/en/gamer, http://dme.rwth-aachen.de/en/SSCG.