2016 IEEE 16th International Conference on Data Mining

Mining Statistically Significant Attribute Associations in Attributed Graphs

Jihwan Lee
Department of Computer Science
Purdue University
West Lafayette, IN
Email: jihwan@purdue.edu

Keehwan Park
Department of Computer Science
Purdue University
West Lafayette, IN
Email: park451@purdue.edu

Sunil Prabhakar
Department of Computer Science
Purdue University
West Lafayette, IN
Email: sunil@purdue.edu

***Abstract*—Graphs are widely used to represent many different kinds of real world data such as social networks, protein-protein interactions, and road networks. In many cases, each node in a graph is associated with a set of its attributes and it is critical to not only consider the link structure of a graph but also use the attribute information to achieve more meaningful results in various graph mining tasks. Most previous works dealing with attributed graphs take into account attribute relationships only between individually connected nodes. However, it should be greatly valuable to find out which sets of attributes are associated with each other and whether or not they are statistically significant over an entire graph. Mining such significant associations, we can uncover novel relationships among the sets of attributes in the graph. We propose an algorithm that can find those attribute associations efficiently and effectively, and show experimental results that confirm the high efficacy of the proposed algorithm.**

I. Introduction

Graphs have emerged as a powerful abstract data type to represent and analyze complex data in a broad range of commercial and scientific applications including social networks, bioinformatics, and the world wide web. Mining structured patterns in graphs has been actively studied in the literature for patterns such as cliques [1], subgraphs [2], paths [3] and trees [4]. When the graph data come with auxiliary information such as node attributes, such information can be applied to various application areas, e.g., community detection, link prediction, graph clustering, network modeling, etc. Thus, attributed graphs are more important than ever before to complex mining tasks.

While node attributes can be successfully employed to augment various mining tasks, the node attributes themselves could give us interesting patterns for understanding graphs better. Given an attributed graph where each node is associated with its attribute values, one might be interested in a pattern of node attribute values which co-occur between connected nodes. Let us call such co-occurring attribute values between two connected nodes an attribute association. This information can tell us directly which attribute patterns are shared by connected nodes over the entire graph. On a large scale, one might be interested in which attribute associations are most frequently observed, or which attribute vector is most expected to be observed given another attribute vector in attribute associations. Even though frequent attribute associations reveal the most dominant attribute associations in the graph, they do not identify which ones are really significant. That is because the frequency of an attribute association often does not depart from what we expect and therefore may not be meaningful if we already know the distributions of attribute values in the graph. Rather, identifying the statistically significant attribute associations where the pattern of the attribute association deviates from the expected, can potentially infer insightful potential relationships between nodes in the graph. The statistical significance of a pattern has been emphasized in various data mining problems [5], [6] and the previous works already explored why a statistically significant pattern is more important rather than a frequent pattern.

The statistical significance of an attribute association with its frequency k is determined by the probability that it is observed at least k times or more, and the probability is called the *p-value* of the attribute association. By measuring *p-value*, we can identify significant associations even though they may not be frequent in the graph. Also, we are interested in even associations of partial attribute values as long as they are statistically significant. The main challenge of the problem is how to estimate the probability that an attribute association occurs in a random graph. If we consider partial attribute associations then the number of possible attribute associations grows exponentially. We address the challenge by transforming a graph G into an alternative graph $\mathcal{AG}$, called *association graph*, where each vertex contains a subset of nodes in G that have the same or similar attribute values and each edge corresponds to a certain attribute association between two sets of attribute values, each of which is represented by a cluster. During the process of transformation, we build $\mathcal{AG}$ such that the edges (i.e., associations) are statistically significant.

In this work, we first formally define the novel problem of mining statistically significant attribute associations which aims to find patterns of co-occurring attribute values between nodes which deviate from the expected. Next, we design and implement an algorithm that can find the statistically significant attribute associations efficiently and effectively. Lastly, we conduct experiments using real world attributed graphs and show qualitative results as well as the actual application that can benefit from the results.

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.154

TABLE I: Basic notations

Notation	Meaning
$G = (V, E, A)$	attributed graph
$\mathcal{AG} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$	association graph
$\vec{a} = (a^1, a^2, \ldots, a^l)$	vector of l number of attributes
Δ	attribute association
σ	*freq_support*
λ	*size_support*
δ_G	density of graph G

II. Problem Statement

A. Attribute Associations

Suppose we have an attributed graph $G = (V, E, A)$ where $V = \{u_1, u_2, \ldots, u_{|V|}\}$ is a set of nodes, $E = \{V \times V\}$ is a set of edges, and $A = \{\vec{a_{u_1}}, \vec{a_{u_2}}, \ldots, \vec{a_{u_{|V|}}}\}$ is a set of attribute vectors, each of which is associated with a node in V. The attribute vector $\vec{a_u}$ of the node u that holds l different attributes is represented by a vector of l binary values in that each binary indicates whether the node u actually has a value for the corresponding attribute (in case of an m multi-valued attribute, it can be transformed into $m-1$ dichotomous binary variables). We define an *attribute association* between a pair of attribute vectors $\vec{a_1}$ and $\vec{a_2}$ as follows:

Definition 1: Given two attribute vectors $\vec{a_1} = (a_1^1, a_1^2, \ldots, a_1^l)$ and $\vec{a_2} = (a_2^1, a_2^2, \ldots, a_2^l)$, the attribute association between them, denoted by $\Delta_{\vec{a_1},\vec{a_2}}$, is defined as a pair of sets of attribute values, $\{i|a_1^i = 1\}$ and $\{i|a_2^i = 1\}$ where $i \in \{1, 2, \ldots, l\}$.

If an attribute association Δ is repeatedly observed and its frequency exceeds a given threshold σ, referred to as *freq_support*, then we say Δ is a frequent attribute association.

Definition 2: Given an attribute association Δ and a support threshold σ, Δ is called a frequent attribute association if $fr(\Delta) \geq \sigma \times |E|$ where $fr(\Delta)$ is the number of pairs of nodes with Δ.

When a frequent attribute association is given, we can say that there are many pairs of nodes having the association but it does not necessarily mean that the attribute association is really interesting. For example, in a social network of *Purdue University Almuni*, it is not surprising to observe many connected nodes with the attribute association of {"Purdue", "CS"} – {"Purdue", "CS"}.

B. Statistically Significance

The statistical significance of an object can be quantified by estimating the probability of the observed or rarer objects under the null hypothesis. Let δ_G denote the density of G which is defined as the fraction of the number of edges in G over all pairs of nodes ($\delta_G = \frac{|E|}{1/2 \cdot |V| \cdot (|V|-1)}$). If we randomly select two groups of nodes no matter which attribute value they have, denoted by C_1 and C_2 respectively, then the number of edges M between C_1 and C_2 would follow the binomial distribution with parameters $n = |C_1| \cdot |C_2|$ and $p = \delta_G$, and thus the probability of getting exactly k edges among n possible edges is given by the following probability mass function:

$$f(k; n, p) = P[M = k] = \binom{n}{k} p^k (1-p)^{n-k} \tag{1}$$

If each of C_1 and C_2 is a group of nodes with the same attribute values in G which are specified by an attribute vector, then the attribute vectors $\vec{a_1}$ and $\vec{a_2}$ can be instantiated from C_1 and C_2 respectively and the attribute association between two attribute vectors is induced from the edges across the nodes of C_1 and the nodes of C_2. So we can measure the statistical significance of a given attribute association $\Delta_{\vec{a_1},\vec{a_2}}$ based on the probability $P[M \geq k]$ and the association is said to be statistically significant if the estimated probability $P[M \geq k]$ is very small.

$$P[M \geq k] = 1 - \sum_{i=0}^{k-1} \binom{n}{i} p^i (1-p)^{n-i} \tag{2}$$

Definition 3: An attribute association $\Delta_{\vec{a_1},\vec{a_2}}$ between C_1 and C_2 is statistically significant if the probability of the observed or more number of edges between C_1 and C_2 is less than α which is called a significance level.

III. Graph Transformation

The basic approach for finding statistically significant attribute associations is to transform the original graph G into a new graph $\mathcal{AG} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, which is called *Association Graph*, where each node in $\mathcal{V}$ corresponds to a group of nodes in V which have the same or similar attribute values, each edge in $\mathcal{E}$ is an attribute association Δ between two attribute vectors, and each attribute vector in $\mathcal{A}$ represents one shared by a group of nodes in V. To avoid confusion, from now on we call a node in $\mathcal{V}$ a cluster and call an edge in $\mathcal{E}$ an association. Each association Δ is assigned a weight, referred to as its strength, that is given by the number of edges between nodes in the clusters forming the association. For a given association Δ and its associated strength, we can determine whether Δ is significant by looking at the strengths and sizes of the clusters to which Δ is incident, as explained in the following sections.

The graph transformation can be done through an iteration of two steps. We first start with a single cluster that contains all nodes of V in G and then the cluster is partitioned into several subclusters by applying two steps repeatedly and iteratively. For the first step, a cluster is split such that each subcluster contains a subset of V that have similar attribute values. This operation is easily performed using any clustering algorithms. In the case of binary attributes, we just select one of the attributes and then do a two-way split with respect to the attribute. In Section III-A, we explain how to select the attribute. For the second step, we try to split a cluster such that each of the associations incident to the cluster has higher strength in order to obtain more significant associations between two sets of attributes. The iteration of these two different splits alternate between performing the similarity-based split, which produces clusters with the same or similar attribute values, and the strength-based split, which makes

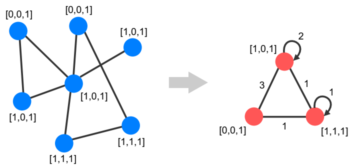


Fig. 1: Graph transformation

associations more significant. It results in a new graph $\mathcal{AG}$ where we can see groups of nodes with certain attribute values and significant associations between them, as shown in Fig. 1.

When obtaining significant associations, each attribute value does not always have to take discrete attribute value, e.g., 0 or 1 in binary case, as long as the association has enough statistical significance. Accordingly, we introduce wildcard attribute notation ($*$), which matches any value of the corresponding attribute.

A. *Similarity-based split*

As mentioned already, the goal of the first step is to maximize the similarity among attribute values in each cluster so that each cluster can represent a certain set of attribute values. Thus, we select one of the clusters in $\mathcal{AG}$ and then split it into two subclusters based on a certain attribute so that each subcluster contains a set of nodes that share the same value on the attribute. The cluster selection in $\mathcal{AG}$ is based on the idea that we want to not only maximize the similarity of attribute values in each subcluster after the split, but also want each subcluster to be statistically significant as much as possible in terms of the attribute values of its nodes.

To achieve the goal, we need to figure out which cluster should be split and which attribute should be used to split that cluster. Let p_i denote the probability that a value of 1 occurs at i-th attribute. First, an attribute on which a cluster should be split is picked such that the probability of the attribute having the value of 1 in the cluster is least deviated from its corresponding p_i. This allows the subclusters to not only have higher similar attribute values among the nodes in them but also have the highest significance gain through the split. Once we decide which attribute should be used for the split, we select which cluster to split. While assuming that the attributes are independent of each other and that the number of times the value of 1 appears at the i-th attribute follows the binomial distribution with the probability p_i, the statistical significance Ψ_c of a cluster c is defined based on the product of p-values of the attribute values of the nodes in the cluster.

$$\Psi_c = 1 - \prod_{i=1}^{l} \left(1 - \sum_{j=0}^{k_i-1} \binom{|c|}{j} p_i^j (1-p_i)^{|c|-j}\right) \tag{3}$$

where k_i is the number of nodes having the value of 1 on the i-th attribute and $|c|$ is the number of nodes in the cluster c. So for each cluster c we compute $\Psi_{c'}$ of the subclusters c'. Remember that our goal is to split a cluster so that its subclusters are most statistically significant. However, since the subclusters may have different significance, we take subclusters with the lowest significance from each of the clusters in $\mathcal{AG}$ and then select a cluster that will produce a subcluster with the highest significance among those subclusters, i.e.,

$$\arg\max_c \left(\min_{c' \in sb(c)} \Psi_{c'} \right) \tag{4}$$

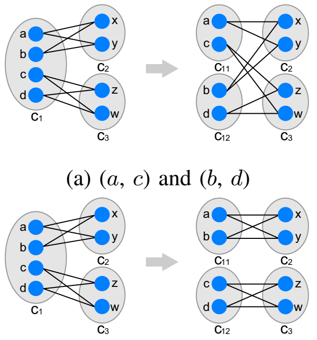
where $sb(c)$ is a set of subclusters that will be created after the split. In this way, we can avoid splitting a cluster that will produce the least significant subclusters. Repeating such split, $\mathcal{AG}$ will have only those clusters for which the same attribute values are shared by its nodes. However, we need to place one constraint while doing the split. Even though a cluster represents a certain set of attribute values shared in it, if it contains only a few nodes then its attribute values may not be meaningful at all when we look at an attribute association between clusters in $\mathcal{AG}$. Thus, we use *size_support*, denoted by λ, to force a cluster not to split any more if all the subclusters that will be obtained after splitting the cluster have the sizes less than $\lambda \cdot |V|$. Thus, during the first step, we examine only clusters satisfying the λ threshold to determine which cluster should be split. Also, it is obvious that a cluster in which all its nodes have the same attribute values does not need to be split.

We do not only want nodes in the same cluster to have the same attribute values but also allow them to have similar attribute values. In other words, even though every node in a cluster does not agree on a certain attribute, if the distribution of the values of the attribute is statistically significantly deviated from the expectation, then those nodes are considered to have an identical value for the attribute.

B. *Strength-based split*

While the similarity-based split of the first step aims to increase the similarity of attribute values for a cluster, we try to maximize strengths of associations to which a cluster is incident through the strength-based split. Given an attribute association between two clusters, its strength is not meaningful by itself because the significance depends on the sizes of the clusters as well as the strength. According to the definition of a statistically significant attribute association, the stronger strength an attribute association has and the smaller the associated clusters are, the more statistically significant the association is. Thus, in order to make an association more significant, a cluster that is one of the end points of the association needs to be split into subclusters such that nodes which have many common neighbor clusters belong to the same subcluster. Fig. 2 illustrates two different splits to maximize the significance of the associations held by a cluster.

Thus we need to find the optimal split of a cluster so that its associations become more significant. To split a cluster c, we build a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ where $\tilde{V} = \{u | u \in c\}$ and $\tilde{E} = \{(u, v) | u, v \in c \wedge \exists c'$ s.t. $(u, w_1), (u, w_2) \in E$ and $w_1, w_2 \in c'\}$, some of which are connected to each other if they have edges with some common neighbor clusters, $\Gamma(c)$. Those edges in $\tilde{E}$ are weighted based on the fraction of edges to common neighbors among all of their edges. Then, we partition the graph $\tilde{G}$ based on the weights of the edges in the graph and the



(a) (a, c) and (b, d)

(b) (a, b) and (c, d)

Fig. 2: Two different strength-based splits: The nodes a and b in c_1 have edges, all of which are incident to other nodes in c_2 while the nodes c and d are adjacent to only other nodes in c_3. Thus, in order for the subclusters obtained from splitting c_1 to have associations of maximized significance, the split should produce two subclusters which contain the two nodes a and b, and the other two nodes c and d, respectively.

subgraphs resulting from the partition become the subclusters we obtain through the strength-based split. For this task, we assign the edge weights by modifying the Jaccard index that is widely used to measure a tie-strength between two nodes, which is given by

$$TS(u, v) = \frac{\sum_{c' \in \Gamma(c)} \min\{\phi(u, c'), \phi(v, c')\}}{\sum_{c' \in \Gamma(c)} \max\{\phi(u, c'), \phi(v, c')\}} \tag{5}$$

where $\phi(u, c') = |w|w \in c \wedge (u, v) \in E|$, that is the number of edges in E between u and any nodes in c'. Using this tie-strength measure, we can have nodes belong to the same subcluster after the split if they have many common neighbor clusters, regardless of whether or not they have common neighbor nodes in G (of course, it depends on the weight given by $TS(\cdot, \cdot)$).

Once we have $\tilde{G}$ for the cluster c, we perform graph partitioning on $\tilde{G}$ to find optimal subclusters that can make the associations between c and $c' \in \Gamma(c)$ more significant. Since all the edges $\tilde{E}$ of $\tilde{G}$ are assigned weights and $\tilde{G}$ should be partitioned based on the weights, we take an approach to maximize the modularity of $\tilde{G}$ [7]. The modularity $Q(\tilde{G})$ is defined as

$$Q(\tilde{G}) = \frac{1}{2m} \sum_{u,v} \left[A_{uv} - \frac{k_u, k_v}{2m} \right] \delta(c_u, c_v) \tag{6}$$

where $m = \tilde{E}$, k_u is the degree of u, c_u is the group to which u belongs, and A_{uv} is 1 if there is an edge in $\tilde{E}$ between u and v otherwise 0. If we split the cluster c through the graph partitioning method as described, a set of nodes that share many common neighbor clusters is likely to fall within the same subcluster as much as possible, and different nodes that share only few common neighbors would be distributed to different subclusters. Thus, we can increase the statistical significance of the attribute associations.

During the second step, we enforce a couple of conditions to prune some clusters and associations in $\mathcal{AG}$ for both achieving computational efficiency and finding more meaningful results.

TABLE II: Dataset statistics

	Original graph			Association graph	
	Nodes	Edges	Density	Nodes	Edges
DBLP	4,672	37,726	0.00346	195	6,302
Yelp	4,454	44,906	0.00453	202	8,388

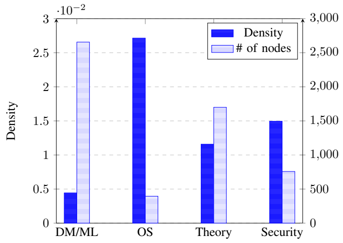


Fig. 3: DBLP subgraph characteristics for different subareas

As done in the first step, the strength-based split is run for a cluster c only when $|c| \geq \lambda \cdot |V|$. In addition to that, if a cluster has an attribute association with low strength, we can safely discard it for the rest of the algorithm. Note that the strength of an attribute association between two clusters monotonically decreases as the two splits are performed iteratively while the statistical significance is not monotonic in either way. Since we consider only attribute associations between clusters satisfying the *size_support* condition and the statistically significance of an association depends on its strength and the sizes of the clusters at the end points, we can prune an attribute association from $\mathcal{AG}$ as long as it meets the following condition.

Lemma 1: Given an attribute association Δ_{c_1,c_2} and its two incident clusters c_1 and c_2, if the strength of Δ_{c_1,c_2} is less than $\Phi^{-1}\left(1 - \alpha - \frac{C(p^2+q^2)}{\sqrt{npq}}\right)\sqrt{npq} + np$, then Δ_{c_1,c_2} does not have a chance to be statistically significant any more, where $n = |c_1| \cdot |c_2|$, $p = \delta_G$, $q = 1 - p$, $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-t^2}{2}}\, dt$, and C is a constant.

Proof: Due to the limit of space, the proof can be found in [8]. ■

According to Lemma 1, we drop attribute associations if they are too weak to become significant later. In fact, such associations are noise and do not add insight. Rather, they prevent the strength-based split from running optimally.

IV. Experiments

We ran the graph transformation algorithm on two real world networks, the DBLP co-authorship network and the Yelp social network, and obtained the resulting association graphs. More details of the data[1] can be found in [8]. Using the association graphs, we analyzed qualitative differences between the statistically significant and frequent associations. Also we showed the application of the significant patterns to a link prediction problem, and the scalability of our algorithm with synthetic attributed graphs.

A. Effectiveness Analysis

To evaluate the effectiveness of our algorithm, we computed the set difference between the statistically significant associ-

[1] available at https://github.com/redrum21c/attribute-associations

TABLE III: Significant associations minus Frequent associations for DBLP

#	Association
1	{SOSP, OSDI, S&P, CCS} – {SOSP, OSDI}
2	{SOSP, OSDI, S&P, CCS} – {S&P, CCS}
3	{ICML, ICDM, S&P(*)} – {ICML, ICDM}
4	{SOSP, OSDI, S&P, CCS} – {SOSP, S&P, CCS}
5	{FOCS(*), STOC(*), CCS} – {S&P, CCS}
6	{ICML, ICDM, S&P(*)} – {ICML(*), ICDM}
7	{SOSP, S&P, CCS} – {SOSP, OSDI}
8	{SOSP, S&P, CCS} – {S&P, CCS}
9	{ICML, ICDM, S&P(*)} – {ICDM, OSDI(*)}
10	{ICML, ICDM} – {ICML(*), ICDM}

ations and the top-15 frequent associations. As the resulting significant associations contain wildcard attributes, it is not easy to make direct comparisons or set differences between the two. Thus, we took the conservative approach that as long as all attribute values of any top-15 frequent associations have exact or wildcard attribute matches, we consider that there is a match. This approach is certainly in favor of the frequent associations, since it ignores that wildcard matches may lead to some other possible set of attribute values.

Table III shows the set difference between statistically significant and frequent associations for the DBLP dataset. First consider 4 subgraphs that only contain the nodes and their edges, whose attribute value for any conference of the corresponding subarea is 1. Fig. 3 describes the characteristics of each subgraph. The subgraph of DM/ML has a large number of nodes but its graph density is small, which means that the tie-strengths are weak. On the other hand, there are relatively small numbers of nodes in the subgraph of OS and security but their densities are high, which means that the tie-strengths are strong among the nodes. We can easily identify that the OS and security-related associations, which contain {SOSP, OSDI} and {S&P, CCS}, appear at the top of the difference list. Also note that many frequent associations are related to DM/ML conferences since its subgraph contains the most number of edges while its density is low.

From Table III, we can infer many interesting significant associations, which do not appear in the frequent association list. The association numbers 1, 2, 4, 7, and 8 clearly show that the nodes that have authorship in the OS-related conferences tend to publish with the authors in the security-related conferences. The association numbers 3, 5 and 6 show that the nodes that have authorship in the security-related conferences frequently publish with the authors in DM/ML and theory-related conferences. Interestingly enough, the association number 9 shows how the authors in DM/ML, security, and OS have frequent co-authorship relations in the graph. These results might look obvious to readers who have a good understanding of co-authorship in computer science. However, when the relationships of attributes are little known, the discussed results may be intriguing.

Table IV shows that the set difference between statistically significant and frequent associations for the Yelp dataset. Note that {Chinese, Japanese} appears very commonly in the association results due to their prevalence in node attributes. Thus, we will exclude them from the subsequent discussion.

TABLE IV: Significant associations minus Frequent associations for Yelp (C: Chinese, J: Japanese, M: Mediterranean, T: Thai, V: Vietnamese, K: Korean, G: Greek, I: Indian)

#	Association
1	{C, J, M, T, G} – {C, M, T(*), G}
2	{C, J, M, T, V, K} – {C, J, M, T, G}
3	{C, J, T, V, K} – {C, J, V, K}
4	{C, J, M, T, V, K, I} – {C, J, M, T, V, K}
5	{C, M, T(*), G} – {C, M}
6	{C, J, M, T, V, K, I} – {C, J, T}
7	{C, J, M, T, V, K} – {C, M, T(*), G}
8	{C, J, M, T, G} – {C, M}
9	{C, J, T, V, K} – {C, J, T}
10	{C, J, M, T, V, K, I} – {C, J, M, T, G}

Also it turned out that the first 10 significant associations with the highest statistical significance are the same as the associations reported in Table IV. That is, none of the first 10 significant associations are reported in the top-15 frequent association results, since the significant associations do not occur often in terms of frequency but do occur often in the dataset in a statistically significant manner.

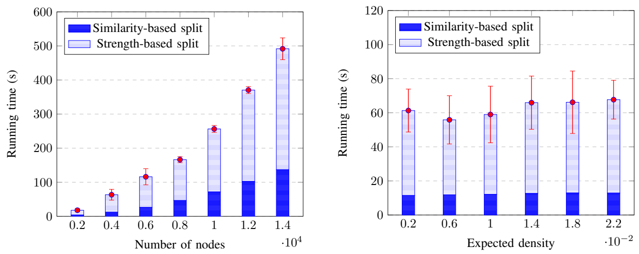
Among the frequent visitors of {Mediterranean, Thai}, the association numbers 2 and 7 show that the nodes with {Greek} attribute are strongly associated with the nodes with {Vietnamese, Korean} attributes; and the association numbers 4, 6 and 10 show that the nodes with {Vietnamese, Korean, Indian} are strongly associated with the nodes with {Vietnamese, Korean} and {Greek} attributes. Also the association numbers 5 and 8 describe that the nodes with {Mediterranean} have statistically significant associations with the nodes with {Mediterranean, Thai, Greek}.

B. Scalability Analysis

We evaluated the computational cost of our algorithm on synthetic attributed graphs of different sizes and densities. The experiments were carried out on a machine with an Intel Xeon 3.1GHz CPU and 32GB memory, running 64bit Ubuntu 14.04. The graphs are generated based on the simplified version of the Multiplicative Attribute Graph (MAG) model [9] with five binary attributes for each node and keeping the node attribute distributions fixed throughout the experiments.

Time complexity. Our algorithm is divisive in nature and it splits at least one node of the *Association Graph* in every iteration. The similarity-based split step will run $\mathcal{O}(2^l)$ iterations. Usually the length of attribute vector is small, $l \ll n$, and the similarity-based split under reasonable settings takes much less time compared to that of the strength-based split. In the strength-based split step, it is not hard to see that the computation of tie-strengths between each pair of nodes, $\mathcal{O}(n^2)$, dominates the running time of the step. And we can notice that the algorithm will run $\log n$ iterations of the strength-based steps on average. Accordingly, the overall average time complexity of the algorithm is $\mathcal{O}(n^2 \log n)$.

Results. Fig. 4a shows the computation time over the number of nodes. We fixed the attribute link-affinity matrix [9], which determines the probability of edge formation between two sets of node attributes. Note that since we fixed all parameters of the MAG model but the number of nodes,



(a) $\mu = 0.6$, density= 0.010

(b) $\mu = 0.6$, nodes= $4,000$

Fig. 4: Running time experiments on synthetic graph datasets

the graph density remained the same. We confirmed that the algorithm is polynomial time in the number of nodes. This result is in line with the time complexity discussed above.

In the second experiment, we fixed the number of nodes and the scale factor of the attribute matrix, which merely changes the expected number of edges. That is, we scaled the attribute matrix such that the resulting graphs have the graph densities as we desire, without changing any other properties of the graphs. In Fig. 4b, we can easily observe that the algorithm's running time remains almost the same as we increase the expected graph density. The aforementioned time complexity should well explain the result.

Finally, both of the plots in Fig. 4 show that the running time of the strength-based split step dominates that of the similarity-based step. Also both plots describe that the running time of the similarity-based step remain the same as we add more edges with the number of nodes fixed, and the running time grows as we increase the number of nodes. This supports our intuition that the similarity-based split step is not relevant to the number of edges or graph density.

C. Application: Link prediction

As one of the applications for which the statistically significant attribute associations are useful, the *link prediction* problem is considered. Many different approaches to the link prediction have been proposed for the past decade, but with the objective of showing the potential merit of the statistically significance attribute associations, we simply use the Jaccard coefficient proposed in [10] and compare the effects of using statistically significant attribute associations and frequent ones. Given a pair of nodes without an edge, we compute the prediction score by combining the Jaccard coefficient $J(u, v)$ and the score $S(u, v)$ resulting from either the significance or the normalized frequency of an attribute association between the nodes as follows

$$pred(u, v) = \tau \cdot J(u, v) + (1 - \tau) \cdot S(u, v) \quad (7)$$

If it is over a given threshold then we predict that u and v will form a new link. We take two snapshots of the *DBLP co-authorship network* (Mar 2015 and Mar 2016) and all the newly created links between the two snapshots are used for the positive samples. Similarly, a set of pairs of nodes that do not have an edge in both the snapshots are used for the negative samples. Since the number of negative samples far outweighs the number of positive samples, we do negative subsampling with the ratio of 1:5 (five negatives per one positive). In Fig. 5, we report the ROC curves for two different methods, `Jaccard+Significant`, and `Jaccard+Frequent`. As shown in Fig. 5, the link prediction can benefit more from employing the attribute information, and the statistically significant attribute associations can achieve higher performance rather than the frequent ones.

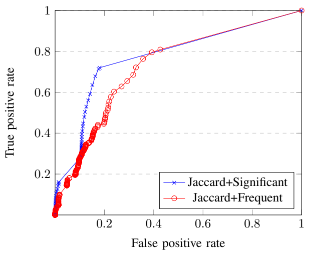


Fig. 5: Link prediction performance

V. Conclusion

We defined a novel problem of mining statistically significant attribute associations using *Association Graph*, which keeps the locality of attribute associations and carries the significant relationships between the sets of attribute values. We proposed a novel, two-step iterative algorithm that efficiently and effectively generates an *Association Graph* from the original graph. Experiments conducted on two real world datasets, and qualitative analysis on the results, confirm that our algorithm effectively finds the significant associations which cannot be uncovered by conventional frequent association mining. We also ran extensive scalability experiments on synthetic datasets, and confirmed that the algorithm is of polynomial running time in the number of nodes. Lastly, applying the results from one of the real world datasets to the link prediction task, we showed how the statistically significant attribute associations can be used in practice.

References

[1] J. Pei, D. Jiang, and A. Zhang, "Mining cross-graph quasi-cliques in gene expression and protein interaction data," in *ICDE 2005*.

[2] S. Ranu and A. K. Singh, "Graphsig: A scalable approach to mining significant subgraphs in large graph databases," in *ICDE 2009*.

[3] J. Scott, T. Ideker, R. M. Karp, and R. Sharan, "Efficient algorithms for detecting signaling pathways in protein interaction networks," *Journal of Computational Biology*, vol. 13, no. 2, pp. 133–144, 2006.

[4] Y. Chi, Y. Yang, and R. R. Muntz, "Indexing and mining free trees," in *ICDM 2003*. IEEE, 2003, pp. 509–512.

[5] H. He and A. K. Singh, "Graphrank: Statistical modeling and mining of significant subgraphs in the feature space," in *ICDM 2016*.

[6] A. Arora, M. Sachan, and A. Bhattacharya, "Mining statistically significant connected subgraphs in vertex labeled graphs," in *SIGMOD 2014*.

[7] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[8] J. Lee, K. Park, and S. Prabhakar, "Mining statistically significany attribute associations in attribute graphs," *CoRR*, vol. abs/1609.08266.

[9] M. Kim and J. Leskovec, "Multiplicative attribute graph model of real-world networks," *Internet Mathematics*, vol. 8, no. 1-2, pp. 113–160, 2012.

[10] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.