# Reaching a desired set of users via different paths: an online advertising technique on a micro-blogging platform

Milad Eftekhar
Department of Computer Science
University of Toronto
Toronto, ON, Canada
milad@cs.toronto.edu

Nick Koudas
Department of Computer Science
University of Toronto
Toronto, ON, Canada
koudas@cs.toronto.edu

Yashar Ganjali
Department of Computer Science
University of Toronto
yganjali@cs.toronto.edu

## ABSTRACT

Social media and micro-blogging platforms have been successful for communication and information exchange enjoying vast number of user participation. Given their millions of users, it is natural that there is a lot of interest for marketing and advertising on these platforms as attested by the introduced advertising platforms on Twitter and Facebook.

In this paper, inspired by micro-blogging advertising platforms, we introduce two problems to aid ad and marketing campaigns. The first problem identifies topics (called *analogous* topics) that have approximately the same audience in a micro-blogging platform as a given query topic. The main idea is that by bidding on an analogous topic instead of the original query topic, we reach approximately the same audience while spending less of our budget. Then, we present algorithms to identify expert users on a given query topic and categorize these experts to finely understand their diversified expertise. This is imperative for word of mouth marketing where individuals have to be targeted precisely.

We evaluate our algorithms and solutions for both problems on a large dataset from Twitter attesting to their efficiency and accuracy compared with alternate approaches.

## Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data mining*; J.4 [**Social and Behavioral Sciences**]: Economics

## Keywords

Social media, Micro-blogging advertising platforms, Analogous topics, Alternative topics, Expert categorization

## 1. INTRODUCTION

Social media and micro-blogging have experienced exponential growth in user acquisition and participation over the last decade. Services such as Twitter, Facebook, and Pinterest allow millions of people to share billions of content and interact on a daily basis. Social platforms are targets of sophisticated advertising and marketing, mainly because of the large number of users, and the enormous amount of time users spend on them.

In micro-blogging platforms (e.g. Twitter), social connections get established by "following" an individual $u$. By establishing such a connection, you get to receive and view all posts (tweets) produced by $u$. The set of all posts that are visible by a user $v$ is commonly referred to as the feed (timeline) of $v$. The act of following someone explicitly expresses interest in the information that person produces.

Social media and micro-blogging platforms are utilized by many as important marketing vehicles. By amassing a large number of followers, an individual or a company can broadcast messages targeted to these followers. Such messages vary depending on the type of the account (e.g., celebrity, professional, consulting, corporate) and what one wishes to achieve (e.g., brand/product awareness, sales leads, or general information dissemination). Typically, one produces information in the field of one's expertise – which is a topic or a set of topics that one knows well, professes, or is known for as an *expert* in the community. For example a celebrity (say a singer) will disseminate information of interest to fans, such as tour dates, personal events and announcements, as well as new songs and albums, whereas a company, say a technology startup, shares information related to its products, and the overall technology product space.

Recently, new advertising platforms have been introduced [10, 17]. In contrast to the keyword bidding model, as is popular in the case of search engine advertising, the micro-blogging platform takes a different approach. An advertiser selects a topic $q$, bids a specific dollar amount, and provides a post (known as a promoted post). The micro-blogging advertising platform identifies all the users that are *interested* in the topic $q$ based on some internal algorithms and inserts the promoted post in the feed of these users (explicitly identifying it as a promoted post). The dollar amount is utilized by an auction that determines the winning bidder (for topic $q$). As an example, if we are interested in showing a promoted post to those users that are interested in *music*, we will bid an amount for the topic *music* and provide our promoted post. If we win, our promoted post will be inserted in the feed of those accounts interested in topic *music*. Commonly the amount we bid is per impression or per engagement (i.e., per person seeing or clicking on the promoted post).

10.5441/002/edbt.2015.17

In such a setting there are numerous opportunities for optimization. Of immediate interest would be reaching the same or approximately the same set of people with a lower cost. For example, by bidding on the topic *public relations* we can be successful only if we bid a price of $x$. What if we knew that if we bid on the topic *seo* (search engine optimization), we can reach the same people (and thus have the same impressions) for a price $y < x$? The first problem we outline in this paper is to produce a set of topics $R$ *analogous* to a topic $q$ (that we wish to bid on). These topics have the property that if we bid on one of them instead of $q$, our promoted post will be inserted in the feed of approximately (for a precise definition of approximate) the same people as those in the case of $q$. Now, by examining the associated cost of each topic, we can make a more informed decision by comparing the savings versus how many interested individuals our posts will reach for each of the analogous topics in $R$. We propose an algorithm called IAT to address this problem (Section 3).

Note that by advertising on a cheaper topic $t \in R$ instead of $q$, (1) (approximately) the same people see the ad and (2) expectedly same people engage with the ad. The cost we should spend in case of targeting $t$ instead of $q$, therefore, would be lower (1) per impression (in cost per impression model) and (2) per engagement (in cost per engagement model). Hence by bidding on $t$, we reach same audience with a lower cost independent of the cost model in use.

Utilizing this technique provides a win-win situation for advertisers and the advertising platforms (e.g., Google, Twitter, etc.). Adopting the technique, advertisers who are interested in a topic will have more options (more topics with similar audience) to target. This prevents from the existence of a very popular topic that is too expensive to target alongside some cheaper topics that no one targets. In this situation, more advertisers afford to advertise. Hence the revenue of the advertising platform may significantly increase while advertisers also obtain more savings per advertisement.

A second popular marketing activity on micro-blogging platforms is to engage *experts* on specific topics into word of mouth marketing campaigns. By having experts on a topic become advocates of a product or a service, all of their followers become informed about the product or service. This is a typical form of word of mouth marketing. For example, if we are interested to market a new cloud computing product by word of mouth on a social media, we can engage cloud computing experts and persuade them to adopt, use, or talk about our product.

Finding the right advocates online is always challenging. Commonly, a user's account has a set of topics associated with it highlighting its expertise. Even if we have an a priori knowledge of the specific topics we wish our advocates to have expertise on, it may be impossible to find one that spans all these topics. Thus, a more iterative approach is desirable. Given that we can identify all experts on a single topic $q$, it would be very useful if we are capable of categorizing those experts based on other topics of their expertise. That would enable us to examine them in a more refined fashion and identify those that are closest to our topics of interest. For example, a set of experts in both *cloud computing* and *virtualization* may be more suitable for us than a set of experts in *cloud computing* and *data centers*. Being able to compute such expert groups algorithmically, given one specific topic of expertise $q$ (cloud computing in our example), is imperative. We have typically no knowledge of what is the "right" number of groups and it is expected that some experts belong to many groups. We propose an algorithm (called CTE, Section 4) to group together all experts on any given topic in a varying number of groups (corresponding to high-level topics) based on the collective topics of expertise of all these users.

The problems discussed in this paper are inspired by social media and micro-blogging advertising platforms. Since the internal algorithms utilized by these platforms are unknown to public, we have proposed some models (e.g., expert identification, topic bidding model, etc. that are explained in the next sections) and utilized them in this paper as a proof of concept. We note that these models, our assumptions, and our methods are *not* based on or designed for any specific social media or micro-blogging platform.

As we have access to a large dataset from Twitter, we evaluate the algorithms on this dataset for various queries. Both IAT and CTE algorithms operate fast (a few minutes) in all experiments stressing the practicality of our developments. In addition, we deploy a qualitative study demonstrating the goodness of our findings and compare our CTE algorithm with some baseline techniques (Section 5). A literature review is provided in Section 6, followed by Section 7 concluding our discussion.

## 2. THE TARGETS

Different social media and micro-blogging platforms such as Twitter, Facebook, and Google+ have introduced the concept of *lists* (*circles* in case of Google+). A list is a user-defined set of accounts. Commonly, users create a list grouping their favourite accounts on a particular topic into that list which they annotate with a descriptive title. For example, in Twitter a user may create a list with the title of "politics" that include Twitter accounts @BarakObama, @AngelaMerkel, @HillaryClinton, @JohnKerry, and @DavidCameron. The utility of a list is to provide quick filtering (by list title) of posts from accounts belonging in the list. It is very typical to group together accounts that profess or depict expertise on a particular topic. A user can create multiple lists and an account can belong to any number of lists.

We utilize the infrastructure of the Peckalytics system [2] to associate with each account $u$, a set of topics $T_u$ extracted from the titles of the lists containing that account. The process of extraction includes tokenization of the title, common word (stop word) and spam filtering, entity extraction, and related word grouping via Wikipedia and WordNet. The end result is, for each account $u$, a set of topics that best describes the topics associated (by other users) with $u$. We emphasize however, that *any* process of mapping an account to a set of topics that best describes the account can be utilized (e.g., machine learning methods). The techniques presented herein will work fine without any modification.

A user $u \in U$ is an *expert* on topic $t \in T$, iff $t \in T_u$. This means that (for our specific way of extracting topics) other users recognize $u$ as an expert on topic $t$. We call topic $t$, a topic of expertise for $u$. The set of experts on topic $t$ is denoted by $E_t$. A user $u \in U$ is *interested* in topic $t \in T$ iff the probability that $u$ follows (reads) any content (a post, a shared video, a posted link, etc.) that is related to topic $t$ is higher than a given threshold $\theta \in [0, 1]$. For a topic $t$, we refer to the set of all users who are interested in $t$ as the *target set* of topic $t$ denoted by $S_t$.

Micro-blogging platforms utilize several factors (content of posts, followers, etc.) to identify the interests of users and subsequently form target sets. However, such factors are largely proprietary. In this paper, we approximate the target set of a topic $t$ by partitioning it into two categories: (1) users interested in $t$ who are also expert on $t$ ($E_t$) and (2) users interested in $t$ who are not expert on $t$ ($I_t$). In other words, users in $E_t$ are producers of contents related to $t$, and users in $I_t$ are consumers of contents related to $t$. Thus, $S_t = E_t \cup I_t$. For any topic $t$, the set of experts $E_t$ is available to us; i.e., $E_t = \{u | t \in T_u\}$. However, the $I_t$ sets are unknown to us (i.e., we do not know which users are interested in a given topic $t$). One may suggest to retrieve the interests for each user by taking the union of expertise topics of all accounts this user follows. This approach has some drawbacks. A given user (say $u$) may be an expert on several topics. When another user (say $v$) follows $u$, the user $v$ may be interested in any of these topics but not necessarily in all of them. It is not straightforward to determine which topic is of interest to $v$, given the topics of $u$'s expertise. In section 3.1, we present an approach to resolve this issue.

# 3. ANALOGOUS TOPICS

In an advertising scenario on a social media platform, by placing a bid for a particular topic $q$, assuming that the bid is granted, users in $S_q$ will observe the promoted post on their feeds. Naturally, an interesting question is whether there is any other topic $t$ that is cheaper than $q$ (i.e., it is possible for a lower bid to be granted) with a target set $S_t$ "close" to $S_q$. If possible, this would reduce advertising cost. This question is the key component of this Section. To formalize the question, we introduce some definitions.

Definition 1. *When a promoted post corresponding to a topic $q$ is shown to a user belonging to $S_t \cap S_q$ (for some topic $t$), we say a* true impression *is achieved. If the promoted post is shown to a user in $S_t \backslash S_q$, we call that a* false impression. *Note that $X \backslash Y$ denotes the set difference between two arbitrary sets $X$ and $Y$. As users in $S_t \backslash S_q$ are not interested in $q$, presenting a promoted post to them is not a desired outcome.*

Definition 2. *The* distance *between two arbitrary sets $X$ and $Y$, denoted by $D(X, Y)$, is the size of the symmetric difference between them: $D(X, Y) = |(X \backslash Y) \cup (Y \backslash X)|$. Moreover, the* distance *between two topics $q$ and $t$ is the distance between their target sets $S_q$ and $S_t$: i.e., $D(q, t) = D(S_q, S_t)$.*

We note that a low distance between topics $q$ and $t$ translates to a high true impression and a low false impression since $D(S_q, S_t) = |(S_t \backslash S_q) \cup (S_q \backslash S_t)| = |S_t \backslash S_q| + |S_q| - |S_t \cap S_q|$. Note that $|S_q|$ is a constant for a fixed query topic $q$.

Definition 3. *A topic $t$ is* analogous *to topic $q$ iff the distance between $q$ and $t$ is less than a given threshold $k \in \mathbb{N}$; i.e., $D(q, t) < k$. That is, $t$ is* analogous *to topic $q$ iff the true impression ($S_t \cap S_q$) is high while the false impression ($S_t \backslash S_q$) is low.*

The goal of this section is to identify a list of topics that are *analogous* to a query topic $q$. These topics are ranked subsequently based on a *weight function* (Equation 10) that involves both true and false impression values. If any of the *analogous* topics has a bidding cost lower than $q$, it is a potential alternative for bidding purposes.

Problem 1. *Let $q$ be a given query topic. Identify all topics $t$ that are* analogous *(Definition 3) to $q$.*

The solution to Problem 1 can be utilized by advertisers to instigate advertising campaigns by choosing the *analogous* topics instead of query topic $q$, target (approximately) the same set of audiences, and pay less.

Problem 1 could be solved if the target sets for all topics were known. Unfortunately, as explained in Section 2, finding the targets sets is not straightforward (since the $I_t$ sets are unknown). To address this problem, in the rest of this section, we present an approach to identify analogous topics without calculating the exact target sets.

## 3.1 Properties of analogous topics

The target set of a topic $t$ can be partitioned into two sets: the set of experts $E_t$ and the set of interested users $I_t$. According to Section 2, the set $E_t$ can be readily identified utilizing the lists. However, $I_t$ is unknown. We aim to identify topics $t$ such that $I_t$ and $I_q$ are "close" (for a suitable definition of close).

We reason about approaches to identify these desired topics. Through this reasoning, we gain some intuition about the properties of analogous topics. Based on the discussion, we conclude this section by introducing two properties of analogous topics that enables us to identify them without calculating the $I_t$ sets.

**Approach I:** A well-known measure of similarity between two arbitrary sets $X$ and $Y$ is the correlation coefficient, denoted by $\rho(X, Y)$. The correlation between two sets can be calculated utilizing the Pearson product-moment correlation coefficient [15]: $\rho(X, Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$ that is equal to $\frac{n(\sum_{i=1}^{n} x_i y_i) - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{(n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2)(n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2)}}$ on a sample, where $n$ is the number of elements, and $x_i$ ($y_i$) is 1 if the $i^{th}$ element belongs to $X$ ($Y$) and 0 otherwise.

In Theorem 1, we show that there exists a direct translation between the correlation coefficient and the distance of two sets.

Theorem 1. *For any two arbitrary sets $X$ and $Y$, if the correlation between them is greater than a threshold $\delta \in [-1, 1]$, there exist a threshold $k \in \mathbb{N}$, negatively associated with $\delta$ ($k \sim -\delta$), such that the distance between $X$ and $Y$ is less than $k$:*

$$\forall X, Y, \forall \delta \in [-1, 1],$$
$$\exists k \in \mathbb{N}, k \sim -\delta, \rho(X, Y) > \delta \Leftrightarrow D(X, Y) < k \qquad (1)$$

Proof Proof Sketch. An increase in $\rho(X, Y)$ is equivalent to an increase in $\sum x_i y_i$ (number of similar items in both sets) that is equivalent to a decrease in $-\sum x_i y_i$ hence a decrease in $D(X, Y)$. Moreover, any increase in $\delta$ and subsequently $\rho(X, Y)$ translates to a decrease in $D(X, Y)$ and subsequently $k$. □

Definition 4. *We define the* correlation *between two arbitrary topics $t$ and $t'$, denoted by $\rho(t, t')$, as the correlation between their target sets; i.e., $\rho(t, t') = \rho(S_t, S_{t'})$. Furthermore, we define the* expertise correlation *between two topics $t$ and $t'$, denoted by $\rho_E(t, t')$, as the correlation between their sets of experts; i.e., $\rho_E(t, t') = \rho(E_t, E_{t'})$.*

According to Theorem 1, for a given query topic $q$, all topics with a high correlation value with $q$ can be reported as the analogous topics. Since the target sets are unknown, the correlation between two topics cannot be computed. However, we can compute the expertise correlation as follows (the Pearson product-moment correlation coefficient):

$$\rho_E(t,t') = \rho(E_t, E_q) = \frac{cov(E_t, E_q)}{\sigma_{E_t}\sigma_{E_q}} = \frac{n(\sum_{i=1}^{n} t_i q_i) - r.s}{\sqrt{(n\sum_{i=1}^{n} t_i^2 - r^2)(n\sum_{i=1}^{n} q_i^2 - s^2)}} \qquad (2)$$

where $n$ is the number of users, and $r$ ($s$) is the number of expert users on topic $t$ ($q$). Moreover, $t_i$ ($q_i$) is 1 if the $i^{th}$ user is expert on topic $t$ ($q$) and 0 otherwise. The denominator is equal to $\sqrt{(nr - r^2)(ns - s^2)}$. The correlation coefficient can vary from -1 (negatively correlated) to +1 (positively correlated).

A basic approach to approximate the correlation between two topics might be to calculate the expertise correlation between them and to utilize it as a metric to assess the correlation between those topics; i.e., one may claim that $\rho(t,q) \sim \rho_E(t,q)$.

Note that a high expertise correlation between $t$ and $q$ suggests that the distance between $E_t$ and $E_q$ is small (Theorem 1). Thus, among experts, the true impression is large and the false impression is small. The primary idea of Approach I is that if the expertise correlation between $t$ and $q$ is high, one may conclude that the correlation between the whole target sets $S_t$ and $S_q$ is high; hence, according to Theorem 1, the distance between $S_t$ and $S_q$ would be small and $t$ would be analogous to $q$ (Definition 3). Unfortunately, this is not correct as clarified by the following example.

Example 1. *Consider two topics "oil" and "Persian classic dance". Note that as Persians actively argue about both topics (suppose independently in separate posts), many users may place them in lists corresponding to each topic. Therefore, many Persians belong to the expertise sets of both topics, creating a high expertise correlation between these topics. In this sense, we may end up concluding that the topic "oil" is analogous to the topic "Persian classic dance". On the other hand, however, the target sets of these two topics can be very different. A person who is interested in "oil" is not necessarily interested in "Persian classic dance". In other words, by targeting the interested users in one of these topics, we do not target the users interested in the other topic. Thus, a high correlation between sets of experts does not imply the same for the corresponding sets of interested users; therefore the target sets for these topics are not necessarily related and $\rho(t,q) \not\sim \rho_E(t,q)$.*

This problem may be resolved by not looking at topics "in isolation" but in conjunction with other topics. The two topics "oil" and "Persian classic dance" have high expertise correlation. However, let us consider other topics with high expertise correlation to "oil" or "Persian classic dance". The topic "oil" has high expertise correlation to topics in $\mathcal{S}_{oil} = \{energy, power, war, \cdots\}$ for example, whereas "Persian classic dance" has high expertise correlation to topics in $\mathcal{S}_{dance} = \{art, music, culture, \cdots\}$. The expertise correlation between topics in $\mathcal{S}_{oil}$ and $\mathcal{S}_{dance}$ is extremely low.

Example 1 suggests that a holistic view, that considers the expertise correlation of all topics in conjunction rather than individual topics in isolation, might help to determine topics that are analogous to $q$. It is a natural tendency of users to be interested in high-level topic categories as well as topics under these categories. For example, if user $u$ is interested in Wimbledon (the tennis tournament), it is natural to assume, that with a high probability, $u$ is interested in the bigger category tennis as well as other tennis events such as French Open, US Open, Australian Open, etc. If $u$ is interested in Oscars, it is safe to assume, with a high probability, $u$ is also interested in other film events such as Golden Globe, BAFTA, Cannes film festival, Berlin film festival, etc. Based on this, we can conclude that for topics in the same category (e.g., Wimbledon and US Open which are both tennis events), the sets of interested users are close (i.e., if $t_1$ and $t_2$ are members of the same category of topics, the distance $D(I_{t_1}, I_{t_2})$ is small). This suggests that identifying the topics in the category that topic $q$ belongs would aid in locating the analogous topics.

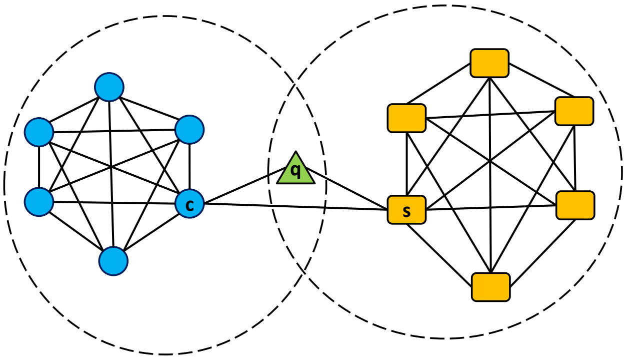


**Figure 1: Partitioning a graph of topics. Any optimal clustering algorithm would generate two clusters as shown: node $q$ may be assigned to each of the two clusters. Note that an optimal clustering will not generate a cluster $\{q, c, s\}$.**

**Approach II:** One approach to incorporate this holistic view might be to calculate the expertise correlation between all topics and create a correlation graph where nodes represent topics and the weight of an edge between two arbitrary nodes $t$ and $t'$ is $\rho_E(t,t')$. Then partition (or classify) this graph and report all topics in the partition containing topic $q$, as the topics analogous to $q$. Unfortunately, this approach has its own shortcomings as shown in the following example.

Example 2. *Consider the graph shown in Figure 1. Each node in this graph represents a topic with node $q$ being the query topic. All the nodes represented by a circle have a high expertise correlation with each other, and all nodes represented by a square have a high expertise correlation with each other. The expertise correlation is small between a circle and a square node. Node $q$ has a high expertise correlation with nodes $c$ and $s$, and there is a high expertise correlation between nodes $c$ and $s$.*

*If we are looking for topics analogous to $q$, ideally one should identify $c$ and $s$. However, any clustering scheme that relies on a global objective function based on the expertise correlations will partition this graph into two clusters as shown in Figure 1 without returning $\{q, c, s\}$ as a separate cluster. We note that for any algorithm that generates a given number of partitions $k$, one can generalize this example by creating sets of $k$ different shapes, without changing the behavior observed.*

The problem encountered in Example 2 is a result of the fact that clustering algorithms rely on the optimization of a global objective function that does not take into account the original topic of interest $q$. Assigning $q$ to a cluster takes place based on the optimality of a global function, and that can lead to poor performance in situations where the focus is solely on $q$.

**Conclusion:** These two examples suggest that one needs a hybrid approach considering both the direct expertise correlation between each topic and $q$, as well as the expertise correlation amongst the neighbors.

A topic $t$ is analogous to topic $q$ if and only if:

PROPERTY 1: The expertise correlation between $q$ and $t$ is greater than a threshold $\delta$; i.e., $\rho_E(q,t) > \delta$.

PROPERTY 2: Topic $t$ is in the same category of topics as topic $q$. In other words, the topics having high expertise correlation with topic $t$ should have high expertise correlation with topic $q$ and vice versa:

$$\forall t'; \rho_E(t,t') > \delta \Leftrightarrow \rho_E(q,t') > \delta$$

When property (1) is satisfied, the distance between $E_t$ and $E_q$ is small (Theorem 1). Moreover, property (2) suggests that the distance between $I_t$ and $I_q$ is small. Therefore, when both properties are satisfied, the distance between the target set of topic $t$ ($S_t = E_t \cup I_t$) and the target set of topic $q$ ($S_q = E_q \cup I_q$) is small; thus $t$ is analogous to $q$.

In most real world scenarios, it is impossible to identify topics $t$ that strictly satisfy both properties. Therefore, we introduce a technique in Section 3.2 that considers both properties, by defining a "trade-off" between them. In other words, our approach assigns weights to any topic $t$ based on the direct expertise correlation between $t$ and $q$ (Property 1), and at the same time penalizes that weight (by associating a cost) if the topics having high expertise correlation with $t$ have low expertise correlation with $q$, or the topics with low expertise correlation with $t$ have high expertise correlation with $q$ (Property 2).

## 3.2 Computing analogous topics

Recall that $U = \{u_1, u_2, \cdots, u_n\}$ denotes the set of users and $T = \{t_1, t_2, \cdots, t_m\}$ denotes the set of topics.

DEFINITION 5. *The* expert coverage probability *of topic $t \in T$, denoted by $\mathcal{P}(t)$, is the fraction of users in $U$ that are expert on $t$. In particular, $\mathcal{P}(t) = \frac{|E_t|}{|U|}$. Moreover, the* expert coverage probability *of topic $t \in T$* given *topic $q$ (the* conditional expert coverage probability, *denoted by $\mathcal{P}(t|q)$) is the fraction of users in $U' = E_q$ that are expert on $t$. In particular, $\mathcal{P}(t|q) = \frac{|E_t \cap E_q|}{|E_q|}$.*

As an example, suppose $U$ consists of 1000 users, among them 10 users are expert on "drawing" and 50 users are expert on "music". This leads to $\mathcal{P}(drawing) = 10/1000 = 0.01$ and $\mathcal{P}(music) = 0.05$.

For any two topics $t$ and $q$, the probabilities $\mathcal{P}(t|q)$ and $\mathcal{P}(t)$ may be significantly different. As an example, assume we observe that among experts on topic "Picasso", 60% are expert on "drawing" and 5% are expert on "music". Thus, $\mathcal{P}(drawing|Picasso) = 0.6 >> P(drawing)$ while $\mathcal{P}(music|Picasso) = \mathcal{P}(music) = 0.05$ (showing that music and Picasso are independent topics). We argue that these changes in the expert coverage probability of different topics given a fixed topic can be utilized as an equivalent measure to Property 1.

We utilize a two-state automaton to study whether any topic $t$ is analogous to a given query topic $q$ or not. This automaton has two states $\mathcal{N}$ and $\mathcal{A}$ corresponding, respectively, to the concepts of "Not-analogous" ($t$ and $q$ are not analogous) and "Analogous" ($t$ and $q$ are analogous). Given topic $q$, while considering topic $t$, the automaton can be in one of the states $\mathcal{N}$ or $\mathcal{A}$. The $\mathcal{N}$ state corresponds to low conditional expert coverage probability and the $\mathcal{A}$ state corresponds to high conditional expert coverage probability. These states determine how far $\mathcal{P}(t|q)$ is from the original $\mathcal{P}(t)$ assessing whether topic $t$ satisfies Property 1 (Theorem 2). For any topic $t$, we aim to identify the state of the automaton with the maximum likelihood.

We deploy a binomial distribution as the basis to realize such measurement. The binomial distribution is a density function that determines the probability that $r$ successes are achieved in a sequence of $d$ independent experiments, when a success is yield with a fixed probability $p$. In the case of topics and experts, this expresses the probability that among $d$ experts on $q$, $r$ users are expert on a topic $t$ where $p = \mathcal{P}(t)$. Adhering to the binomial distribution, the probability that the automaton is in state $\mathcal{N}$ for a topic $t \in T$ is:

$$\mathcal{P}(\mathcal{N}_t|q) = \frac{\binom{d}{r_t}\mathcal{P}(t)^{r_t}(1-\mathcal{P}(t))^{d-r_t}}{Z} \tag{3}$$

where $r_t = |E_t \cap E_q|$, $d = |E_q|$. Similarly the probability that the automaton is in state $\mathcal{A}$ is:

$$\mathcal{P}(\mathcal{A}_t|q) = \frac{\binom{d}{r_t}(\alpha \times \mathcal{P}(t))^{r_t}(1-\alpha \times \mathcal{P}(t))^{d-r_t}}{Z} \tag{4}$$

where $\alpha > 1$ (a constant), and $\alpha \times \mathcal{P}(t)$ is the expected expert coverage probability of $t$ given $q$ in case $t$ is analogous to $q$. Here, $Z = \binom{d}{r_t}\mathcal{P}(t)^{r_t}(1-\mathcal{P}(t))^{d-r_t} + \binom{d}{r_t}(\alpha \times \mathcal{P}(t))^{r_t}(1-\alpha \times \mathcal{P}(t))^{d-r_t}$ is a normalizing constant. Since the denominator $Z$ is similar in both equations and does not impact the calculations, hereafter, we ignore it and just consider the numerators in calculating and comparing $\mathcal{P}(\mathcal{A}_t|q)$ and $\mathcal{P}(\mathcal{N}_t|q)$.

THEOREM 2. *The value of $\frac{\mathcal{P}(\mathcal{A}_t|q)}{\mathcal{P}(\mathcal{N}_t|q)}$ increases (decreases) iff the distance $D(E_t, E_q)$ decreases (increases).*

PROOF. Let $\alpha$ be fixed. The value of $\frac{\mathcal{P}(\mathcal{A}_t|q)}{\mathcal{P}(\mathcal{N}_t|q)}$ increases (decreases) when $\alpha^{r_t}(\frac{1-\alpha\mathcal{P}(t)}{1-\mathcal{P}(t)})^{d-r_t}$ increases (decreases). The latter increases (decreases) when $r_t$ increases (decreases) or $\mathcal{P}(t)$ decreases (increases). This is because $\alpha > 1$ and $0 < \frac{1-\alpha\mathcal{P}(t)}{1-\mathcal{P}(t)} < 1$. Moreover $\mathcal{P}(t)$ decreases (increases) when $|E_t|$ decreases (increases). In both cases $D(E_t, E_q)$ decreases (increases). □

To incorporate Property 2, we create a *correlation graph*: a graph $G = (M, E)$ where any topic $t \in T - \{q\}$ corresponds to a node in $G$. Moreover, for any two nodes $m_i, m_j \in M$ representing topics $t_i$ and $t_j$, the weight of the edge $e$ connecting $m_i$ and $m_j$ is $w_e = w(m_i, m_j) = \rho_E(t_i, t_j)$.

DEFINITION 6. *Suppose the state of the automaton for any topic is determined. In particular, the automaton is in state $s_i$ ($\mathcal{N}$ or $\mathcal{A}$) when considering topic $t_i$ ($t_i$ corresponds to node $m_i$ in $G$). The edge $e = (m_i, m_j)$ is called* inconsistent *if ($w_e > 0$ and $s_i \neq s_j$) or ($w_e < 0$ and $s_i = s_j$).*

Definition 6 suggests that an edge $e = (m_i, m_j)$ is *inconsistent* if the automaton is in different states when the expertise correlation between topics $t_i$ and $t_j$ (corresponding to nodes $m_i$ and $m_j$) is positive ($\rho_E(t_i, t_j) > 0$) or when the automaton is in the same state if $\rho_E(t_i, t_j) < 0$.

Problem 2 utilizes the automaton and the correlation graph to identify the most likely states for all topics maximizing $\prod_{t_i \in T-\{q\}} \mathcal{P}(s_i|q)$ (or equivalently $\sum_{t_i \in T-\{q\}} \log \mathcal{P}(s_i|q)$) where $s_i \in \{\mathcal{N}, \mathcal{A}\}$ is the state assigned to topic $t_i$. To satisfy Property 2, Problem 2 associates a cost with any inconsistent edge.

PROBLEM 2. *Let $G = (M, E)$ be the correlation graph for topics in $T - \{q\}$. Identify the state of the automaton for each node $m_i \in M$ ($s_i \in \{\mathcal{N}, \mathcal{A}\}$) to*

$$\textit{maximize} \sum_{m_i \in M} \log \mathcal{P}(s_i|q) - \sum_{e \in E} c_e \qquad (5)$$

*Note that $c_e$ is the cost of edge $e = (m_i, m_j)$ that is equal to $|w_e|$ if $e$ is inconsistent or zero otherwise. By adding a constant factor $\sum_{\substack{e \in E \\ w_e < 0}} w_e$ to Equation 5, we get*

$$(5) \equiv \textit{maximize} \sum_{m_i \in M} \log \mathcal{P}(s_i|q) \; - \sum_{\substack{e=(m_i,m_j) \in E \\ s_i \neq s_j}} w_e \qquad (6)$$

Maximizing Equations 5 and 6 is equivalent to maximizing the probability that the correlation graph $G$ is created by a two-state automaton where the probability that the automaton is in state $\mathcal{N}$ or $\mathcal{A}$ for each node in $G$ is derived by Equations 3 and 4 (corresponding to Property 1) and for each edge $e$ in $G$, the probability that the automaton maintains the same state over the two end-points of that edge depends on $w_e$ (corresponding to Property 2).

THEOREM 3. *Problem 2 is NP-hard.*

PROOF. We reduce the max-cut problem to Problem 2. In max-cut problem, given a weighted graph $G$, the goal is to partition vertices of $G$ into two subsets $S_1$ and $S_2$ such that the weight of edges between $S_1$ and $S_2$ is maximized. The max-cut problem is widely known to be NP-hard [12].

The reduction is as follows. Let us assume we want to identify the maximum cut for the graph $G$. We create a graph $G'$ where there is a node $u'$ in $G'$ for any node $u$ in $G$. For any edge $e = (u_i, u_j) \in G$, we add an edge $e' = (u'_i, u'_j)$ in $G'$ ($u'_i$ and $u'_j$ are the nodes in $G'$ corresponding to $u_i$ and $u_j$ in $G$) with a weight of $w_{e'} = -w_e$ where $w_e$ is the weight of edge $e$ in graph $G$. Moreover, for any two nodes $v_i$ and $v_j$ in $G$ not connected to each other, we add an edge between their corresponding nodes $v'_i$ and $v'_j$ in $G'$ with a weight of $w_{e'} = 0$. Finally, we set the probability that the automaton is in the $\mathcal{N}$ or $\mathcal{A}$ state for any node in $G'$ to be equal. Identifying the maximum cut for graph $G$ reduces to solving Problem 2 for graph $G'$ by identifying the state of automaton for any node in $G'$ that maximizes Equation 6 that is equivalent to maximizing $W = \sum_{e'=(u'_i,u'_j) \in E'; s_{u'_i} \neq s_{u'_j}} (-w_{e'})$.

After identifying the optimal states of the automaton for each topic, we define the set $S_1$ containing all nodes in $G$ corresponding to nodes in $G'$ that are assigned with a $\mathcal{N}$ state and $S_2$ containing all nodes in $G$ corresponding to nodes in $G'$ that are assigned with a $\mathcal{A}$ state. Thus, $W = \sum_{e \in E'} w_e$ where $E'$ contains all edges in $E$ with an endpoint in $S_1$ and the other endpoint in $S_2$. In this sense, maximizing $W$ is equivalent to identifying the maximum cut. □

To identify analogous topics, a more general approach would be to model this process by a 3-state automaton. The automaton, for any topic, can be in any of the 3 states "Dissimilar", "Independent", or "Analogous". The conditional expert coverage probability of topic $t$ on these states is, respectively, $a_1$, $a_2$ and $a_3$ where $a_1 < a_2 < a_3$. In particular, $a_2 = \mathcal{P}(t)$ is the expert coverage probability for topic $t$ over $U$, $a_1$ is a lower conditional expert coverage probability showing that the topics $t$ and $q$ are dissimilar (perhaps negatively analogous), and $a_3$ is a higher conditional expert coverage probability showing that the topics $t$ and $q$ are analogous. In the present work, we simplify the model and merge the "Dissimilar" and "Independent" states to form a "Not-analogous" state $\mathcal{N}$. This generalization can be conducted easily following the developments in the section.

## 3.3 IAT: an algorithm to Identify Analogous Topics

According to Theorem 3, Problem 2 is NP-hard. It involves two parts: (1) maximizing the log-likelihood of expert coverage probabilities over all nodes; i.e., $\sum_{m_i \in M} \log \mathcal{P}(s_i|q)$, and (2) minimizing the cost of inconsistent edges; i.e., $\sum_{e \in E} c_e$. The value for the first part can be calculated for each node independently. Computing the second part, however, needs to be aware of the states of the neighboring nodes.

We propose a technique (called IAT) that adopts a heuristic approach to reduce the complexity of Problem 2. The main root of the complexity in Problem 2 is the existence of cycles in the graph. In an acyclic graph, we can order nodes of the graph and identify the best state assignment by optimizing both parts of Equation 5 node-to-node based on this ordering. However, when the graph contains a cycle, no ordering can be assumed between nodes in the cycle; the states of all these nodes depend on each other (due to the second part) and should be determined simultaneously. This leads to a complex structure to deal with. The basic idea of IAT is to obtain an acyclic subgraph (a spanning tree) of the original graph. We, then, identify the optimal states based on this tree. Our experiments show that by utilizing this technique, we can effectively locate the analogous topics.

This approach raises the question on how to choose the spanning tree. In Problem 2, before determining the state of a node, we consider the states of the neighboring nodes in order to reduce the cost of inconsistent edges. Among all edges connected to an arbitrary node $u$, some have the highest probability to be inconsistent. We refer to these as *inconsistency-prone* edges. The goal is to assign the states such that (1) the log-likelihood of expert coverage probabilities over all nodes is maximized and (2) the cost of inconsistency-prone edges is minimized. The idea is that since the inconsistency-prone edges are the most likely edges to induce costs, a state assignment that reduces the cost over these edges, reduces the cost over all edges.

To locate the inconsistency-prone edges, we define an *expected cost* value for each edge. The edges with high expected cost values are considered inconsistency-prone. Let $\hat{A}_u = \frac{\log \mathcal{P}(\mathcal{A}_u|t)}{\log \mathcal{P}(\mathcal{A}_u|t) + \log \mathcal{P}(\mathcal{N}_u|t)}$ determine the expected probability that $u$ is associated with state $\mathcal{A}$ and $\hat{N}_u = 1 - \hat{A}_u$ be the expected probability that $u$ is associated with state $\mathcal{N}$.

The *expected cost* of an edge $e = (u, v)$ denoted by $\hat{c}_e$ is:

$$\hat{c}_e = \begin{cases} |\hat{A}_u - \hat{A}_v| \times w_e & \text{if } w_e \geq 0, \\ (1 - |\hat{A}_u - \hat{A}_v|) \times |w_e| & \text{if } w_e < 0. \end{cases} \tag{7}$$

where the value $|\hat{A}_u - \hat{A}_v|$ (a value between 0 and 1) determines the difference in probability of being associated with state $\mathcal{A}$ for adjacent nodes $u$ and $v$. High values of $|\hat{A}_u - \hat{A}_v|$ suggest that $u$ and $v$ are likely to be assigned with different states. Having the expected cost values, Problem 3 identifies the optimal acyclic subgraph.

Problem 3. *Considering the expected cost of each edge in the graph $G = (M, E)$, identify an acyclic subgraph $T = (M, E^*)$ with maximum sum of the expected costs over all edges in $E^*$.*

Problem 3 is equivalent to the minimum spanning tree problem. We can create a new graph $G'$ by negating the weights of all edges in $G$ and identifying the minimum spanning tree in $G'$. This tree would be the optimal solution for Problem 3 that can be found utilizing any MST algorithm such as Kruskal [11] or Prim [4]. The run time complexity for these algorithms on a dense graph is $O(m^2)$ where $m$ is the cardinality of $M$.

Assume tree $T$ is the optimal solution for Problem 3. The IAT algorithm is a dynamic programming approach that calculates two values $LP(\mathcal{N}_u)$ and $LP(\mathcal{A}_u)$ for any node $u \in M$ starting from leaves, going upwards to the root. For each leaf $u$, $LP(\mathcal{N}_u) = \log \mathcal{P}(\mathcal{N}_u|q)$ and $LP(\mathcal{A}_u) = \log \mathcal{P}(\mathcal{A}_u|q)$ that are calculated based on Equations 3-4. For any inner node $u$, the values are calculated as follows:

$$LP(\mathcal{A}_u) = \log \mathcal{P}(\mathcal{A}_u|q) + \sum_{v \in C(u)} \max(LP(\mathcal{A}_v), LP(\mathcal{N}_v) - w(u, v)), \tag{8}$$

$$LP(\mathcal{N}_u) = \log \mathcal{P}(\mathcal{N}_u|q) + \sum_{v \in C(u)} \max(LP(\mathcal{A}_v) - w(u, v), LP(\mathcal{N}_v)), \tag{9}$$

where $C(u)$ is the set of $u$'s children and $w(u, v)$ is the weight of edge $(u, v)$.

When all values are calculated, IAT identifies the best state assignment to all nodes by locating the chain of states maximizing the value of $\max(LP(\mathcal{A}_r), LP(\mathcal{N}_r))$ where $r$ is the root of the $T$.

The pseudo-code for IAT is presented as Algorithm 1. Note that $p_u$ is the parent of $u$ and $C(u)$ is the set of $u$'s children in Tree $T$. The variable $APointer_u$ ($Npointer_u$) saves the optimal state assigned to $u$ when its parent $p_u$ is assigned with a state $\mathcal{A}$ ($\mathcal{N}$). The function "arg max$(a, b)$" returns $\mathcal{A}$ if $a > b$ and returns $\mathcal{N}$ otherwise. Finally, $s_u$ holds the assigned state of node $u$. IAT reports all topics that are assigned with state $\mathcal{A}$ as the analogous topics. For each analogous topic $t$, we define a weight as

$$weight(t) = \log \mathcal{P}(\mathcal{A}_t|q) - \log \mathcal{P}(\mathcal{N}_t|q) \tag{10}$$

This weight determines the improvement we achieve when topic $q$ is assigned with state $\mathcal{A}$ instead of $\mathcal{N}$. Thus, topics with higher weights correspond to more prominent relationships with topic $q$. Algorithm IAT ranks the analogous topics based on these weight values and returns $Q$, a ranked list of all nodes assigned with a $\mathcal{A}$ state.

**Algorithm 1:** The IAT algorithm

```
   input  : The correlation graph $G = (M, E)$, Topic $q$
   output: A ranked list of analogous topics $Q$
   // Identify the optimal spanning tree
1  Calculate the expected cost of edges according to Eq. 7
2  Identify the optimal spanning tree $T$ (e.g., by Prim)
   // Probability calculations
3  Traverse $T$ bottom-up (from leaves to the root):
4  foreach $u \in M$ do
5      $LP(\mathcal{A}_u) = \log \mathcal{P}(\mathcal{A}_u|q) + \sum_{v \in C(u)} \max(LP(\mathcal{A}_v), LP(\mathcal{N}_v) - w(u, v))$
6      $LP(\mathcal{N}_u) = \log \mathcal{P}(\mathcal{N}_u|q) + \sum_{v \in C(u)} \max(LP(\mathcal{A}_v) - w(u, v), LP(\mathcal{N}_v))$
7      $APointer_u = \arg\max(LP(\mathcal{A}_u), LP(\mathcal{N}_u) - w(u, p_u))$
8      $NPointer_u = \arg\max(LP(\mathcal{A}_u) - w(u, p_u), LP(\mathcal{N}_v))$
   // Identifying the state of the root
9  $s_r = \arg\max(LP(\mathcal{A}_r), LP(\mathcal{N}_r))$
   // Identifying the state for all nodes
10 Traverse $\mathcal{F}$ top-down (from roots to leaves):
11 foreach $u \in M$ do
12     $s_u = NPointer_u$;
13     if $s_{p_u}$ = "$\mathcal{A}$" then
14         $s_u = APointer_u$
   // Sort the analogous topics
15 $Q = \emptyset$
16 foreach $u \in M$ do
17     if $s_u$ = "$\mathcal{A}$" then
18         $Q = Q \cup \{u\}$
19 Sort $Q$ based on Eq. 10
```

Theorem 4. *The IAT algorithm identifies the optimal state assignment on the tree. The run time complexity of IAT is $\theta(m^2)$ where $m$ is the number of topics.*

Proof Proof Sketch. IAT is a standard dynamic programming approach that solves Problem 2 step by step from leaves to the root. The value $LP(\mathcal{A}_u)$ is the optimal solution for Problem 2 on the subtree rooted at $u$ when the state of $u$ is $\mathcal{A}$. Similarly $LP(\mathcal{N}_u)$ is the optimal solution for Problem 2 on the same subtree when the state of $u$ is $\mathcal{N}$. Therefore the value of $\max(LP(\mathcal{A}_r), LP(\mathcal{N}_r))$ is the optimal solution for the whole tree.

We can calculate the expected cost of each edge in constant time. Since there are $m^2$ edges, line 1 takes $\Theta(m^2)$. Prim's algorithm implemented with Fibonacci heap takes $\Theta(m^2)$ to identify the MST. The probability calculation phase takes $\Theta(m)$ since each edge in the MST can update $LP$ of one node only once. The state identification phase also takes $\Theta(m)$ to calculate the optimal states for all nodes. Finally it takes $\Theta(m \log m)$ to sort the analogous topics. Thus, in total IAT takes $\Theta(m^2)$. □

## 4. CATEGORIZING THE FOLLOWERS

In Section 1 we explained it is very helpful to categorize all experts on a given topic $q$ based on other topics of their expertise in order to engage them in word of mouth campaigns. For example, among all experts on *social media*, those who are expert on topics such as "consumer behavior", "distribution channel", "market-based pricing", "sales", etc. can be potentially categorized together (in a big category of "Marketing"); and those expert on topics such as "high ranking placement", "website visitors", "Google results", "search engine traffic", "white hat seo", etc. can form a big category of "Search Engine Optimization".

By categorizing the experts, we would be able to understand them in a more refined fashion and to locate the experts that are the "right" advocates to instigate a popularity propagation (based on word of mouth effects) in the network.

## 4.1 CTE: an algorithm to Categorize Topics and Experts

Let $q$ be a topic, $E_q$ be the set of experts on $q$, and $T_u$ be the set of all topics user $u$ is an expert on. We propose an algorithm (called CTE) to categorize users $u \in E_q$ based on the topics of their expertise. We introduce four desirable properties that CTE should have:
*(1) Soft clustering:* Users may be assigned to several categories. This is desirable as users usually have diverse topics of expertise hence they might belong to various categories.
*(2) Unknown number of categories k:* The optimal number of categories is unknown. The algorithm should identify the best number of categories instead of requiring it as an input.
*(3) Coping with high dimensional data:* The number of topics is large. On high dimensional datasets, any approach based on distance (e.g., the traditional clustering algorithms) is inaccurate since distances between all pairs converge.
*(4) Considering the correlation between topics:* Topics are correlated; any approach that is based on an assumption that dimensions (topics in this case) are independent is not applicable.

In Sections 5 and 6, we argue that traditional clustering algorithms fail to provide useful categorizations. Here, we present an approach (satisfying the properties above) that considers topics and users in two steps: first it categorizes the topics without taking into account the users (topic categorization phase); and then it assigns each user $u \in E_q$ based on $T_u$, to the topic categories (user assignment phase).

This separation of topics and users in categorization helps to segment topics into partitions that are representing high-level topic categories. When we utilize an approach that simultaneously categorizes users in $E_q$ and topics in $\bigcup_{u \in E_q} T_u$ (e.g., the bi-clustering techniques), topics are categorized according to the correlations calculated utilizing the sets $T_u$ of users $u$ in $E_q$, instead of utilizing the sets $T_u$ of all users in $U$. Incorporating the users in $E_q$ (instead of *all* users) to capture correlations introduces *coverage bias*.

Coverage bias loosely means that users in $E_q$ are not representative of the population. There are cases where the correlation between two topics $t_1$ and $t_2$ is low but the topics are highly correlated in the context of a query topic $q$ (i.e., based on users in $E_q$). For example, consider topics "Queen's park" and "Government" in the context of topic "Ontario". These topics are not highly correlated in general. However, when the set we consider consists of experts on "Ontario", the two topics would be highly correlated, since Queen's park is the home for the Legislative Assembly of Ontario and is usually utilized as a metonym for the Government of Ontario. On the other hand, there are cases where two topics $t_1$ and $t_2$ are highly correlated but when considered in the context of experts on a query topic $q$, this correlation is small. Consider two topics "football" and "rugby" given the query topic "fifa" as an example. In general rugby and football are correlated due to the relation between rugby and the American football. However, given the topic "fifa", the term "football" would usually refer to the international "football" that has low correlation with "rugby".

### 4.1.1 *Topic categorization*

The CTE algorithm runs in two phases: (1) topic categorization, and (2) user assignment. Topic categorization starts by creating the correlation graph among topics as discussed in Section 3.2 (incorporating all users in $U$ in weight calculations). Subsequently, we aim to segment topics (graph nodes) into categories such that topics with positive expertise correlation values are located in the same category and topics with negative expertise correlation values are located in different categories.

PROBLEM 4. *Let $G = (V, E)$ be a* correlation graph *where topic $t \in \bigcup_{u \in E_q} T_u$ corresponds to a node in $V$. Also, the weight of the edge connecting any pair of nodes $u_i, u_j \in V$ (representing topics $t_i$ and $t_j$) is $w_{u_i u_j} = \rho_E(t_i, t_j)$. Segment $G$ into categories such that the sum of the weights of edges with positive weights that are cut and edges with negative weights that are uncut is minimized.*

Bansal et. al. have shown that Problem 4 is NP-hard even for a simple case where the weight of all edges are either $-1$ or $+1$ [1]. Demaine et. al. have shown that Problem 4 and the weighted multicut problem are equivalent; Problem 4 is APX-hard; and obtaining any approximation bound better than $\theta(\log n)$ is difficult ($n = |V|$). Utilizing the linear programming rounding and "region growing" techniques, they have proposed an algorithm to approximate Problem 4 with a tight bound of $\theta(\log n)$ [5].

This approach models the problem as a linear program. A zero-one variable $x_{uv}$ is defined for any pair of vertices $u$ and $v$. The equation $x_{uv} = 0$ suggests that $u$ and $v$ are in the same category; $x_{uv} = 1$ declares the opposite. Problem 4 translates to

$$\textit{minimize} \quad \sum_{(u,v):\ w_{uv}<0} |w_{uv}|(1 - x_{uv}) + \sum_{(u,v):\ w_{uv}>0} |w_{uv}| x_{uv}$$

subject to the following constraints:
(1) $x_{uv} \in [0, 1]$, (2) $x_{uv} = x_{vu}$, and (3) $x_{uv} + x_{vw} \geq x_{uw}$.
A "region growing" technique is adopted, afterwards, to transform the fractional values of $x_{uv}$ to integral values 0 or 1. The basic idea is to grow balls around graph nodes (with a fixed maximum radius). Each ball is reported as a category. Therefore, two nodes $u$ and $v$ with a high value $x_{uv}$ would be assigned to two different balls and finally two different categories (equivalent to setting $x_{uv} = 1$).

The run time complexity of the algorithm proposed by Demaine et. al. is $\mathcal{O}(n^7)$. This approach is not practical for datasets containing a large number of topics. In our implementation of topics based on Twitter lists, we construct millions of topics for Twitter users. Any approach based on an $\mathcal{O}(n^7)$ algorithm is deemed not practical for our setting.

We propose a heuristic approach called MaxMerge to categorize the correlation graph when the graph is large. The CTE algorithm utilizes MaxMerge in the topic categorization phase. To start, MaxMerge constructs a category for each vertex in $G$. The algorithm proceeds iteratively. In each iteration, it calculates the value $\Delta_{AB}$ achieved by merging any pair of existing categories $A$ and $B$. The value $\Delta_{AB}$ is the average of the weights of edges with one end-point in category $A$ and one end-point in category $B$. According to Problem 4, the objective is to categorize $G$ such that the edges with positive weights are in the same category and the edges with the negative weights are amongst different categories. The $\Delta_{AB}$ value expresses our progress towards the objective when the two categories are merged. At each iteration, categories $A$ and $B$ having the maximum positive value of $\Delta_{AB}$ will be merged; MaxMerge continues as long as this maximum positive value is greater than the average weight of all node pairs in the whole graph (stating that merging the two categories at hand should result in a value that is higher than the average weight of one big category that includes all topics). Pseudo code of MaxMerge is provided as Alg 2. The input is a graph $G = (V, E)$.

**Algorithm 2:** The MAXMERGE algorithm

```
1 Let $avr$ be the average of the weight of all edges in $E$
2 Consider each node as a category
3 foreach pair of categories $A$ and $B$ do
4     $SUM = \sum$ weight of all edges between A and B
      $\Delta_{AB} = SUM/(|A| * |B|)$
5 $Max = \max_{A,B} \Delta_{AB}$;   $A^*, B^* = \arg\max_{A,B} \Delta_{AB}$
6 if $Max > avr$ then
7     Merge $A^*$ and $B^*$ to one category and Goto step 3
8 return all categories
```

Theorem 5. *The run time complexity of Algorithm 2 is $\mathcal{O}(m^2 \log m)$ where $m = |V|$.*

Proof. Line 1 takes $\mathcal{O}(m^2)$ since there are $m^2$ edges between the topics. Line 2 takes $\mathcal{O}(m)$. At the beginning there are $\mathcal{O}(m^2)$ pairs of partitions and it takes $\mathcal{O}(1)$ to calculate $\Delta$ values for each pair. Thus it take $\mathcal{O}(m^2)$ to calculate these values at the first iteration. We can store these values in a priority queue. Based on the implementation it takes $\mathcal{O}(m^2)$ or $\mathcal{O}(m^2 \log m)$ to create this priority queue.

We do several iterations while $Max > avr$ to merge the partitions. The number of iterations is at most $m - 1$. In each iteration, it takes $\mathcal{O}(\log m)$ to find and delete the max value, $\mathcal{O}(m)$ to merge the two partitions, $\mathcal{O}(m)$ to update the values of $\Delta$ for the new merged partition, and $\mathcal{O}(m \log m)$ to update these values in the priority queue. Note that if we merge two partitions $A$ and $B$ into the new partition $C$, for any partition $D$: $SUM_{CD} = SUM_{AD} + SUM_{BD}$.

Therefore, overall the run time is $\mathcal{O}(m^2 \log m)$. □

### 4.1.2 Assigning the experts

Once the topic categories are identified, we assign users in $E_q$ to these categories. CTE assigns a user $u \in E_q$ based on $T_u$ (Algorithm 3): it assigns $u$ to any category containing at least one topic in $T_u$. Note that adhering to this approach, a user $u$ can be a member of several partitions expressing $u$'s diversified expertise on various high-level topics.

**Algorithm 3:** Expert Assignments

```
Create an empty category $\tilde{C}$ for each topical category $C$
foreach user $u$ in $E_q$ do
    foreach topic category $C$ do
        if there exist topic $t \in C$ such that $t \in T_u$ then
            $\tilde{C} = \tilde{C} \cup \{u\}$
Output all categories $\tilde{C}$
```

## 5. EXPERIMENTS

We evaluate the proposed algorithms IAT and CTE on a dataset containing about 4.5 million lists (that is all lists available in Twitter when we collected the data). Each list $l_i$ is associated with a topic $t_i$ (hence 4.5 million topics)[1] For each user $u$ in a list $l_i$, the corresponding topic $t_i$ is considered as a topic of expertise for $u$ (i.e., $t_i \in T_u$). There are 13.5 million distinct users in these lists.

**Table 1: The Impact of pruning on the number of topics.**

| Query topic | number of experts | number of topics | **size**: number of topics after the 1% pruning |
|---|---|---|---|
| social+media | 375809 | 1360060 | 551 |
| canada+politics | 460 | 19080 | 1490 |
| wine+Toronto | 1337 | 27061 | 938 |
| cloud+computing | 56 | 2769 | 2769 |
| fashion+trends | 1112 | 36263 | 886 |

We execute the algorithms on a machine with a 16 core AMD *Opteron*$^{TM}$ 850 Processor. This machine runs CentOS 5.5 (kernel version 2.6.18-194.11.1.e15) and contains 100GB of memory. All algorithms are single-threaded and are implemented in Java.

We observe similar trends when evaluating our algorithms with different query topics. Here, we report results for the following 5 queries: (1) canada+politics, (2) cloud+computing, (3) social+media, (4) toronto+wine, and (5) fashion+trends. For each query $q$ (e.g., social+media), we retrieve all users whose topics of expertise match each input query (e.g., all users who are expert on social and also on media). These users form the set of experts $E_q$ for the given query $q$.

## 5.1 Identifying the analogous topics

Identifying the analogous topics for the aforementioned queries involves two steps: (1) creating the correlation graph, and (2) assigning $\mathcal{A}$ or $\mathcal{N}$ states to topics (Algorithm IAT).

Figure 2(a) shows the distribution of topics and experts for query social+media. We count how many users in $E_q$ (for a given query $q$) are expert in each topic in $\bigcup_{u \in E_q} T_u$. We see a similar trend for all other queries. This figure suggests that the curve displaying the number of experts on each topic has a heavy tail. Thus, pruning the topics with very small frequency can significantly help in improving performance. The run time of identifying analogous topics for the query social+media is measured utilizing different pruning percentages and reported in Figure 2(b). Here, pruning with a percentage of $\alpha$ means that the topics that appear in the expertise sets of less than $\alpha$% of the experts are removed; i.e., a topic $t$ is pruned iff $\frac{|\{u \in E_q | t \in T_u\}|}{|E_q|} < \frac{\alpha}{100}$.

We see that a pruning of only $1 - 2\%$ can significantly decrease the run time. On the other hand, pruning does not have a major impact on the accuracy of the results. The

[1] The precise process we follow to make this association can be found in [2].

**Table 2: Analogous topics (topics are presented stemmed)**

| | social media | canada politics | toronto wine | cloud computing | fashion trends |
|---|---|---|---|---|---|
| 1 | busi | polit | food | tech | fashion |
| 2 | entrepreneur | news | food wine | cloud | trendsett |
| 3 | pr | canada | foodi | cloud comput | blogger |
| 4 | polit | politicsdemocraci economi | food drink | technolog | blog |
| 5 | journalist | canadian polit | restaur | a68 | fashion blogger |
| 6 | seo | media | canada | cloudcomput | fashionista |
| 7 | entertain | news polit | wine | cloudyp | fashion beauti |
| 8 | info | cdnpoli | toronto food | tech news | media |
| 9 | internet market | politico | canadian | cloud 0 | design |
| 10 | communic | peopl | eat | cloud virtual | fashion blog |
| 11 | industri | canadian | toronto restaur | virtual | shop |
| 12 | advertis | local | chef restaur | news | news |
| 13 | fav | progress | media | busi | creativ |
| 14 | brand | toronto | resto | vendor | lifestyl |
| 15 | engag | journalist | food toronto | techi | fashion style |
| 16 | communiti | blogger | toronto foodi | work | beauti |
| 17 | inform | interest | we like eat drink | softwar | beauti fashion |
| 18 | onlin market | cdn polit | all | cloud saa | busi |
| 19 | digit market | liber | ontario | cloudcomputingenthusiast | inspir |
| 20 | cultur | govern | culinari | clouderati | entertain |

topics that are reported as analogous are very similar for all these pruning percentages (e.g., no difference exists in the top 10 topics when we prune the topics with various percentages 0.1%-10%). We observe similar behavior for all other queries. For the rest of this section, we use a pruning of 1% of the topics to improve performance. Table 1 shows the number of topics that are not pruned in this step for the given five queries. According to Table 1, the pruning step with even a very small value of 1% significantly reduces the dimensionality of the problem. Hence, the problem can be solved more efficiently. The only exception here is the query for cloud+computing. We note that this query has 56 experts. A pruning of 1% removes any topic appearing in the expertise set of less than 0.56 users; thus, no topic is removed in this case (all topics appear in the expertise set of at least 1 user).

Table 2 reports the top-20 topics for each query as identified by IAT. The analogous topics are sorted based on Equation 10. In Table 2, we observe, for example, that topics such as "busi(ness)" (topics are presented stemmed), "entrepreneur", "journalist", "seo", "internet market(ing)", and "communic(ation)" are analogous to the query social+media.

The utility of this information is evident: instead of focusing on topics such as social+media for advertising campaigns (which due to their popularity could involve a high monetary premium), one can focus on peripheral topics, not as popular, but still be able to target an audience close to that of the original query.

Running IAT takes about 0.1 seconds on average for queries evaluated. Note that the majority of the run time to identify analogous topics is taken by the first step. The total run time is bound by the time required to calculate the correlation between the topics.

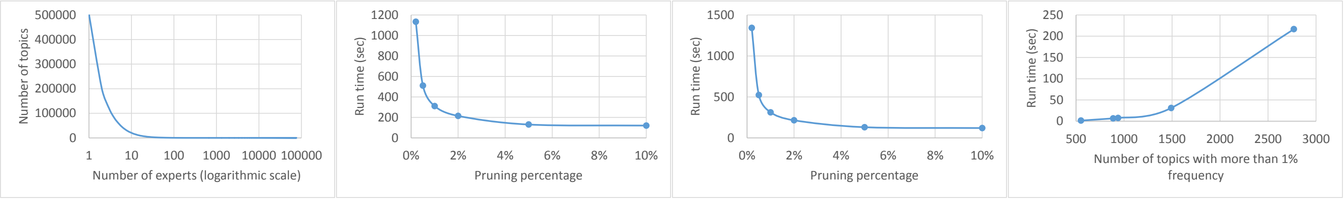
## 5.2 Categorizing experts and topics

The experts and topics categorization is done in two steps: (1) creating the correlation graph, and (2) executing CTE. As Figure 2(c) shows pruning significantly reduces the run time here as well for query social+media. We report the results when a pruning percentage of 1% is utilized. Similar behavior is observed for the other cases. We define the *size* of a query as the number of topics in $\bigcup_{u \in E_q} T_u$ after pruning takes place. Figure 2(d) reports the run time of CTE versus the size for different queries. On average, CTE takes less than 1 minute to run, making CTE practical for most real settings. As Theorem 5 suggests, the run time of CTE increases polynomially when the size increases.

We have evaluated the CTE algorithm for many queries and observed similar trends in results of all experiments. In what follows, due to space constraints, we present results using the query social+media. We stress however that these results are typical and consistent across a wide range of queries we experimented with. Thus, the specific query social+media is representative of the results obtained with algorithm CTE. Table 3 presents the categories identified by CTE for social+media. In each case, we insert an explanation for the set of topics in each category (in bold). It is evident that the contents of each category are highly related; i.e. from that point of view the results do make sense.

Evaluating the output of CTE qualitatively is challenging. To assess the utility of the results of CTE, we need to compare it with other applicable approaches and most importantly obtain confidence that the categories identified are indeed the correct ones. In the absence of ground truth to objectively compare the CTE approach with other applicable approaches (such as clustering), we resort to develop a base reference set that is manually constructed and compare our results against the base set. Running the CTE algorithm on multiple manually created base sets leads to highly consistent results.

To create these base sets, we choose several topics, categorize each topic into subsets pre-selected by us, and manually annotate each set with a descriptive name. Hereafter, we call these new datasets, the *manually annotated datasets*. These manually annotated datasets, present a "ground truth" in which we know (or expect) a preset number of categories to appear. The goal is to categorize topics and users in the manually annotated datasets utilizing different algorithms (without taking the manual annotations into account) and



(a) topic-expert distribution (b) Identifying Analogous topics (c) Categorizing experts (d) CTE vs. size

**Figure 2: Dataset distribution and run time analysis**

**Table 3: Topic categories for the query "social+media". Rows represent categories including a description followed by the topics in each category.**

| |
|---|
| **Tourism in North America** (calgari, ottawa, vancouver, toronto, chicago, san fransisco, seattle; hotel, tourism, travel, beer, wine, restaurant, food, ...) |
| **Australia** (melbourn, sydney, australia, aussi) |
| **UK** (manchester, europe, uk, london) |
| **Sports** (tennis, golf, hockey, baseball, nfl, sport, football) |
| **Health** (mental heath, health well, pharmacy, healthcare, doctor, medic, psychology, ...) |
| **Education** (edu, edtech, learn, university, science, research, academy, ...) |
| **Investments** (invest, economia, economy, financ, realest, realtor, real estat) |
| **South by South "SXSW" festivals** (sxsw, west texas, austin, houston, dallas) |
| **Law** (legal, law, lawyer) |
| **Twibes: groups of people with common interests** (twibe socialnetwork, twibe journal, twibe blog, twibe writer, twibe travel, twibe photographi, twibe webdesign, twibe internetmarket, twibe brand, twibe socialmedia, twibe advertis, twibe entrepreneur, ...) |

compare how close the results are to the manual annotations. We compare CTE with the baseline clustering algorithm k-means (denoted by kmeans in Figure 3) and 3 baseline co-clustering algorithms Euclidean distance (denoted by cocluster Euclidean), Information theoretic (denoted by cocluster IT), and minimum sum-squared residue co-clustering (denoted by cocluster MSR) [3,6,7]. The following categories form one sample manually annotated dataset:
(1) A category $S_1$ including topics {physics, math, chemistry}. We call this category, SCIENCE.
(2) A category $S_2$ including topics {democrats, republican, politics}. We call this category, POLITICS.
(3) A category $S_3$ including topics {soccer, football, fifa}. We call this category, SPORTS.
(4) A category $S_4$ including topics {google, tablet, android}. We call this category, TECHNOLOGY.
We create a dataset $\mathcal{D}$. The set of topics in $\mathcal{D}$ is $S = \{$physics, math, chemistry, democrats, republican, politics, soccer, football, fifa, google, tablet, android$\}$. Users in $\mathcal{D}$ are all users who are expert in at least one topic in $S$ ($U' = \{u \in U | T_u \cap S \neq \emptyset\}$ where $U$ denotes the set of all users in the Twitter dataset). For each user $u \in U'$, $T_u \cap S$ is the set of all topics (among topics in $\mathcal{D}$) that $u$ is an expert on. We compare the results of CTE and the baseline algorithms when deployed to categorize users and topics in $\mathcal{D}$.

The optimal categorization of $\mathcal{D}$ is achieved when (1) the topics in $S$ are categorized into 4 categories $S_1$, $S_2$, $S_3$, and $S_4$ (in accordance with the way the data set was constructed); and (2) the users in $U'$ are categorized into 4 categories of {users who are expert on a topic in $S_1$}, $\cdots$, {users who are expert on a topic in $S_4$}.

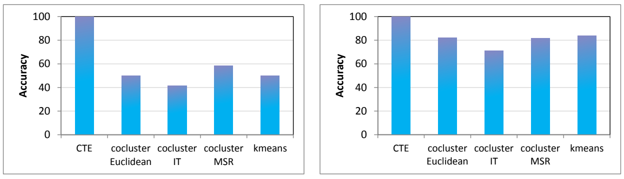
Although algorithm CTE does not need a number of categories as input, the baseline clustering techniques do require the number of clusters (categories). Thus, we provide them with the optimal number 4 providing them with an advantage. Algorithm CTE identifies the optimal number of categories without receiving it as an input.

To calculate the accuracy of an algorithm, we proceed as follows. Assume algorithm X outputs topic categories $C_1, C_2, \cdots, C_r$ and user categories $D_1, D_2, \cdots, D_s$. We utilize four annotations SCIENCE, POLITICS, SPORTS, and TECHNOLOGY to label each category produced by X. A category $C_i$ ($D_i$) is labeled by the annotation having the maximum number of entities in that category. For example, consider a topic category $C_1 = \{$physics, soccer, math$\}$. This category includes two topics in SCIENCE, one topic in SPORTS, and no topic in POLITICS or TECHNOLOGY. Thus, we label the category $C_1$ as SCIENCE. Moreover, assume $D_1 = \{u_1, u_2, u_3, u_4\}$, where $T_{u_1} = \{$soccer, math$\}$, $T_{u_2} = \{$soccer$\}$, $T_{u_3} = \{$math, democrats$\}$, and $T_{u_4} = \{$chemistry, republican$\}$. We can observe that 3 users in $D_1$ are experts on SCIENCE, 2 users on SPORTS, and 2 users on POLITICS. Therefore, we label the category $D_1$ as SCIENCE.

The topic categorization accuracy (user categorization accuracy) of an algorithm is the percentage of the topics (users) that are labeled correctly. Note that in the previous example, one topic is labeled inaccurately in $C_1$ (the topic soccer is labeled as SCIENCE) and one user is labeled inaccurately in $D_1$ ($u_2$ is labeled as SCIENCE without having any expertise on physics, math, or chemistry). Figure 3(a) reports the accuracy for topic categories and Figure 3(b) shows the accuracy for user categories for all algorithms. Figure 3 demonstrates the superiority of CTE when compared with baseline clustering algorithms.

## 6. RELATED WORKS

Twitter lists are recently used to address a few questions such as identifying users' topics of expertise [2,16] and separating elite users (e.g., celebrities) from ordinary users [18]. The problem of identifying a set of topics that can be utilized as a substitute for an expensive topic is studied for the case that target sets of topics are given and the cost for each topic is known [9]. In many real settings we don't have ac-



(a) Topic categories (b) User categories

**Figure 3: Comparison between the accuracy of different clustering algorithms**

cess to this information. This paper focuses on the problem when the target sets and costs are unknown.

Automatons are utilized in several problems such as identifying bursts of activity in time-series data [13], spatial datasets [14], and subgraphs of social networks' graphs [8]. Perhaps the most similar work to our IAT algorithm is the DIBA algorithm [8] that is proposed to identify the bursty subgraphs of users in a social network when the information burst happens as a result of an external activity (such as an earthquake). We note that there are major differences between our IAT and DIBA algorithms: (1) DIBA is mainly designed for unweighted graphs; (2) DIBA does not consider negative edges. In fact, the optimization problem (Problem 2) in presence of negative edges is NP-hard (Theorem 3) while if all weights are non-negative, the problem would become equivalent to min-cut and can be solved in polynomial time [8]; (3) IAT addresses Problem 2 by locating the optimal cycle-free subgraph, while DIBA utilizes a heuristic approach that randomly orders graph nodes and attempts to find the best label for each node in this order; this approach does not identify the optimal subgraph and may ignore considering several important (costly) edges.

The traditional clustering algorithms can be categorized to partitioning methods (e.g., k-means), hierarchical methods (top-down, bottom-up), Density-based (e.g., DBSCAN), model-based (EM), link-based, bi-clustering, and graph partitioning (e.g., finding cliques or quasi-cliques in the graph, and correlation clustering). These algorithms also suffer from several disadvantages in the case of our problem. To the best of our knowledge none of these clustering algorithms provide all of the four desirable properties (introduced in Section 4.1); hence they are not applicable to categorize experts. For completeness, we compared our proposed algorithm with some of these algorithms in Section 5.

## 7. CONCLUSION AND FUTURE WORKS

In this paper we introduce two problems. The first problem is to identify topics (called analogous) that have (approximately) the same audience on a micro-blogging platforms as a query topic. The idea is that by bidding on an analogous topic instead of the original query topic, we will reach (approximately) the same audience while spending less budget on advertising. This is inspired by the social media advertising platforms. The second problem is to understand the diversified expertise of the experts on the given query topic and categorize these experts. We evaluate the techniques proposed for both problems on a large dataset from Twitter attesting their efficiency and accuracy.

An important direction for future work is to study the problems when the bids on each topic is known. This extension can assist advertisers to maximize their revenue while minimizing the advertising cost.

## 8. ACKNOWLEDGMENTS

The authors would like to thank Alex Cheng for his assistance with data preparation.

## 9. REFERENCES

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3):89–113, June 2004.

[2] A. Cheng, N. Bansal, and N. Koudas. Peckalytics: Analyzing experts and interests on twitter. SIGMOD Demo Track, 2013.

[3] H. Cho, I. Dhillon, Y. Guan, and S. Sra. Minimum sum-squared residue co-clustering of gene expression data. SDM, pages 114–125, 2004.

[4] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 3rd edition, 2009. Chapter 23.

[5] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2):172–187, Sept. 2006.

[6] I. Dhillon and Y. Guan. Information theoretic clustering of sparse co-occurrence data. ICDM, pages 517–520, 2003.

[7] I. S. Dhillon, S. Mallela, and D. S. Modha. Information theoretic co-clustering. SIGKDD, pages 89–98, 2003.

[8] M. Eftekhar, N. Koudas, and Y. Ganjali. Bursty subgraphs in social networks. WSDM, pages 213–222, 2013.

[9] M. Eftekhar, S. Thirumuruganathan, G. Das, and N. Koudas. Price trade-offs in social media advertising. In *Proceedings of the Second Edition of the ACM Conference on Online Social Networks*, COSN '14, pages 169–176, New York, NY, USA, 2014. ACM.

[10] Facebook. Facebook help centre. `https://www.facebook.com/help/ads`.

[11] J. Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, June 1956.

[12] R. Karp. Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008*, pages 219–241. Springer Berlin Heidelberg, 2010.

[13] J. Kleinberg. Bursty and hierarchical structure in streams. SIGKDD, pages 91–101, 2002.

[14] M. Mathioudakis, N. Bansal, and N. Koudas. Identifying, attributing and describing spatial bursts. *VLDB Endowment*, 3(1-2):1091–1102, Sept. 2010.

[15] J. L. Rodgers and A. W. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[16] N. K. Sharma, S. Ghosh, F. Benevenuto, N. Ganguly, and K. Gummadi. Inferring who-is-who in the twitter social network. In *Proceedings of the 2012 ACM workshop on Workshop on online social networks*, pages 55–60, 2012.

[17] Twitter. Start Advertising | Twitter for Business. `https://business.twitter.com/start-advertising`.

[18] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 705–714, 2011.