The TagAdvisor: Luring the Lurkers to Review Web Items

Azade Nazi
Department of Computer Science and Engineering
University of Texas, Arlington
azade.nazi@mavs.uta.edu

Mahashweta Das
HP Labs
Palo Alto, USA
mahashweta.das@hp.com

Gautam Das
Department of Computer Science and Engineering
University of Texas, Arlington
gdas@cse.uta.edu

ABSTRACT

The increasing popularity and widespread use of online review sites over the past decade has motivated businesses of all types to possess an expansive arsenal of user feedback (preferably positive) in order to mark their reputation and presence in the Web. Though a significant proportion of purchasing decisions today are driven by average numeric scores (e.g., movie rating in IMDB), detailed reviews are critical for activities such as buying an expensive digital SLR camera, reserving a vacation package, etc. Since writing a detailed review for a product (or, a service) is usually time-consuming and may not offer any incentive, the number of useful reviews available in the Web is far from many. The corpus of reviews available at our disposal for making informed decisions also suffers from spam and misleading content, typographical and grammatical errors, etc. In this paper, we address the problem of how to engage the lurkers (i.e., people who read reviews but never take time and effort to write one) to participate and write online reviews by systematically simplifying the reviewing task. Given a user and an item that she wants to review, the task is to identify the top-f meaningful phrases (i.e., tags) from the set of all tags (i.e., available user feedback for items) that, when advised, would help her review an item easily. We refer to it as the TagAdvisor problem, and formulate it as a general-constrained optimization goal. Our framework is centered around three measures - relevance (i.e., how well the result set of tags describes an item to a user), coverage (i.e., how well the result set of tags covers the different aspects of an item), and polarity (i.e., how well sentiment is attached to the result set of tags) in order to help a user review an item satisfactorily. By adopting different definitions of coverage, we identify two concrete problem instances that enable a wide range of real-world scenarios. We show that these problems are NP-hard and develop practical algorithms with theoretical bounds to solve them efficiently. We conduct detailed experiments on synthetic and real data crawled from the web to validate the utility of our problem and effectiveness of our solutions.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; D.2.8 [Social Media]: Algorithm, Data Mining

Keywords

Personalized Tag Advisor; Relevance; Coverage; Polarity

1. INTRODUCTION

Motivation: The increasing popularity and widespread use of online reviews in sites like Yelp, Amazon, Angie’s List, TripAdvisor, etc. over the past decade has motivated businesses of all types to possess an expansive arsenal of user feedback (preferably positive) in order to mark their reputation and presence in the Web. User feedback is available in various forms such as numeric or star ratings, number of visits, number of check-ins, number of Facebook likes, tags, reviews, etc. Though a significant proportion of purchasing decisions today are driven by aggregate user feedback in the form of average rating (e.g., a movie in IMDB), number of Facebook check-in (e.g., a restaurant page in Facebook), number of views (e.g., an article in Business Insider), etc., detailed reviews continue to influence a wide variety of critical activities such as buying an expensive digital SLR camera, choosing a car, reserving a vacation package, etc. However, since writing a detailed review for a product (or, a service) is usually time-consuming and may not offer any incentive, the number of useful reviews available is far from many. Though the 1% rule (or, the 90-9-1 rule) of Internet is presumed to be dead, the proportion of lurkers (i.e., people who read user-generated content in the Web without contributing) is still high. According to survey conducted by Pew Internet in 2012, though 90% people conduct online product research, only 37% people have ever rated a product, service, and only 32% have ever posted a review online about product they bought or service they received. In addition, several sites like Hotels.com and IMDb allow users to submit feedback as ratings without any review accompaniment. As a result, the number of numerical ratings available for a product far exceeds the number of detailed reviews. The corpus of reviews available at our disposal for making informed decisions suffers from redundancy, inaccurate and misleading content, typographical and grammatical errors, etc. too.

Our Problem: In this paper, we investigate how to engage the users to participate and write online reviews by systematically simplifying the web item (e.g., electronic products, apparel, restaurant, movies, music, travel itineraries, etc.) reviewing task. Given user feedback for items by past users in the form of text, a user and an item that she wants to review, the task is to identify the top-f meaningful phrases (i.e., tags) that we advise to the user in order to help her review the item. We refer to this as the TagAdvisor problem.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGMOD’15, May 31–June 4, 2015, Melbourne, Victoria, Australia.
Copyright © 2015 ACM 978-1-4503-2758-9/15/05 ...$15.00.
http://dx.doi.org/10.1145/2723372.2749457.

∗Work done as graduate student at University of Texas, Arlington

∗ Work done as graduate student at University of Texas, Arlington.
problem. The user would quickly choose from among the set of returned tags to articulate her feedback for the item without having to spend a lot of time writing the review.

The top-\(k\) tags should not only meet the necessary requisites of a good online review like conciseness, comprehensiveness, objectiveness, etc. but should also offer adequate incentive in the form of simple usability, easy applicability, etc. As one of our first step towards solution, we employ state-of-art text mining techniques (discussed later in Section 5) and extract meaningful phrases or tags from user feedback in the form of text, i.e., reviews. Since each tag is a user feedback for an item, the tags are extracted with sentiment labels attached to it. \(T^+\) and \(T^-\) are the set of positive and negative tags respectively. For example, a review statement “It is a lightweight camera with some amazing features” is reduced to the tags [lightweight camera, amazing features], where both tags have positive sentiment.

We formulate the problem of identifying the top-\(k\) meaningful tags from the set of all tags (i.e., available user feedback for items) for a user-item pair, as a novel general-constrained optimization problem. A core challenge in this design is defining the essential properties of the top-\(k\) tags to be returned that would serve to review the item effectively. We consider relevance (i.e., how well the result set of tags describes an item to a user), coverage (i.e., how well the result set of tags covers the diverse aspects of an item), and polarity (i.e., how well sentiment is attached to the result set of tags) in order to enable a user to satisfactorily review an item. Though relevance and coverage have been studied in the past [4], our work is the first to consider all three measures simultaneously in the context of tag mining, to the best of our knowledge.

A user can review an item in different ways. A user can express her broad opinion about the different aspects of an item which, in turn, can either be positive or negative. Again, a user can express both positive and negative opinion for the same attribute (or, set of attributes) of the item. For example, a user may write a review for a camera as “the picture quality of this camera is great and so is the sharpness and color accuracy of the pictures, but the battery life is short.”, while another user of the same camera may write “Though the extra screen with touchscreen and gesture-control features saps battery life, its perfect for fashion-conscious snap shooters.”. The first review contains positive feedback for the camera’s image quality and negative feedback for the camera’s battery life. The second review contains both positive and negative feedback for the camera’s advanced features [dual-screen, touchscreen and gesture-control]. Therefore, the item attributes that were covered by the review is independent of the feedback sentiment in the former case, and dependent on the sentiment in the latter. This motivates us to propose two problem instances, namely Independent Coverage TagAdvisor problem and Dependent Coverage TagAdvisor problem that considers two different definitions of coverage respectively in order to satisfy users’ real world needs.

Related Work: Though the output of our problem is recommending a set of tags for a user-item pair, our objective is different from the literature of work dedicated to tag recommendation [10, 25]. The top-\(k\) tags in our problem are more feedback than descriptive relevant information for an item and hence calls for additional properties like coverage of all aspects of the item in order to ensure diversity, as well as sentiment polarity in opinion of the user for the different aspects of the item. While review summarization, that helps users read the valuable content in the vast volumes of user feedback for items, has been researched in the literature [11, 14, 16, 26], our objective of simplifying a user’s review writing task has not been studied to the best of our knowledge. Moreover, none of the existing work on review summarization, ranking, and selection accommodate all three measures —relevance, coverage, and polarity —that we consider in our framework. Even collaborative filtering based approaches for tag recommendation consider only relevance measure to determine the top-\(k\) tags [15]. We discuss some of these related work in more details in Section 6.

Technical Challenges and Solutions: The TagAdvisor (TA) problem is technically challenging for several reasons. Our objective is to identify \(k\) tags that are relevant, cover different aspects of an item, and have well-balanced positive and negative sentiment attached to it. While the first two concern the relationship between the item attributes and tags, the third is dependent on a user’s personal preference. Some users tend to be lenient and provide mostly positive feedback; some tend to be critical. In this paper, we choose to focus on modeling the complex dependencies that exist between item attributes and tags and leverage user personal preference as a parameter, thereby letting the system deal with both new users and with new items, alleviating cold-start problems. Classifiers and rule learning techniques in the literature can be used to predict the relevance of tags for an item. In this paper, we employ existing techniques to predict the rules modeling the relationship between attributes and tags, where each rule has a probability of occurrence.

As discussed earlier, formalizing the users’ different ways of reviewing an item relates to the coverage characteristic of the top-\(k\) tags to be returned. By adopting different definitions of coverage, we propose problems that enable a wide range of real-world scenarios. For a user reviewing an item, the Independent Coverage TagAdvisor (IC-TA) problem identifies top-\(k\) tags that are relevant, satisfy the user’s criticalness in reviewing, and maximizes the number of item attributes covered by them, independent of their sentiment. On the other hand, the Dependent Coverage TagAdvisor (DC-TA) problem returns tags that cover item attributes both positively and negatively, in addition to being relevant and satisfying user’s criticalness in reviewing. As one of our first results, we show that each of these problem is NP-Complete by reduction from Max-Coverage problem with Group Budget Constraints problem and MAX-SUM Facility Dispersion problem respectively. Given this intractability result, designing efficient algorithmic solutions that work well in practice is challenging. In addition, the objective function of the second problem is proved to be not sub-modular thereby precluding the direct use of off-the-shelf greedy algorithms. For each problem, we first discuss an exact solution (E-IC-TA and E-DC-TA) and then develop an advanced approximation algorithm (A-IC-TA and A-DC-TA) respectively. We prove that each of our approximation algorithm produces solution with constant approximation factor. We conduct experiments on synthetic data and real data crawled from Yahoo! Autos, Walmart and Google Product to evaluate the efficiency and quality of our proposed algorithms. We also present an Amazon Mechanical Turk user study and an interesting case study on real camera data to validate the effectiveness of our solution over that by state-of-art.

In summary, we make the following main contributions:

- We introduce and motivate the novel TagAdvisor problem that leverages available user feedback for items in online review sites to simplify the review writing task. Our objective is to identify the top-\(k\) meaningful tags that, when advised to a user, would help her review an item easily.
- We formulate the problem as a general-constrained optimization goal. Our formulation is centered around three measures —relevance, coverage, and polarity.
- We formalize the users’ different ways of reviewing an item by proposing two coverage functions and thereby defining
two concrete problem instances, namely Independent Coverage TagAdvisor (IC-TA) and Dependent Coverage TagAdvisor (DC-TA) problems, that enable a wide range of real-world scenarios.

- We show that each of the problems is NP-Complete and develop practical algorithms with compelling theoretical properties to solve them efficiently.
- We perform detailed experiments on synthetic and real data crawled from the web to demonstrate the utility of our problem and effectiveness of our algorithms.

2. THE TagAdvisor FRAMEWORK

2.1 Preliminaries

We model the data $D$ in an online review site as a triple $<U,I,T>$, representing the sets of users, items, and the tag vocabulary respectively. Let $n$ be the total number of tags in $T$. Each tagging action can be considered as a triple itself, represented as $<u,i,T>$ where $u \in U$, $i \in I$, and $T \in T$. We assume that each user $u \in U$ has a well-defined schema $U_A = \{c_1, c_2, \ldots\}$, where the attributes typically are the demographic information such as name, age, gender, location, etc. A user is represented as a tuple $<c_1, c_2, \ldots>$ conforming to $U_A$, where $c_i$ is the value of the user attribute $c_i$. Similarly, a camera is represented as $<name=Amy, age=23, gender=Female, location=California>$ represents a 23 years old female from California. Similarly, every item $i \in I$ is associated with a well-defined schema $I_A = \{a_1, a_2, \ldots, a_m\}$ and each item $i$ is a tuple $\{a_{i1}, a_{i2}, \ldots, a_{im}\}$ with $I_A$ as schema, where $a_{ij}$ is the value of item attribute $a_i$; e.g., $<brand=Samsung, model=TL225, type=point and shoot>$ describes a compact Samsung camera. Note that, our work is not influenced by or biased towards any brand. Since each tag is a user feedback for an item, it describes the item positively or negatively. Therefore, we partition $T$ into $T^+$ and $T^-$, where $|T^+|$ is $n^+$ and $|T^-|$ is $n^-$. For example, suppose, we would like to help a user review a camera, say Samsung TL225. Table 1 describes the data available in an online review site where users Amy and David have left tag-based feedback for the camera. Table 1 also shows the attribute values for the users and the camera. The set of all tags $T = \{T_1, T_2\}$ for item $i$ (i.e., Samsung TL225) by users $u_1$ (i.e., Amy) and user $u_2$ (i.e., David) is classified into $T^+ = \{\text{super cool, stylish, lightweight}\}$ and $T^- = \{\text{blurry pictures, gimmicky touchscreen, poor battery life}\}$ by domain experts.

Given an item $i$ and set of tags $T$, probabilistic classifiers can be used to compute the relevance of the tags for the item (i.e., $Pr(t_x|i)$). In this paper, we use the rule based classifiers [6, 17] to find the dependency of the item attributes to the tags and generate rules with probability of occurrence $p$, i.e., the relevance score. However, there exist a number of prior work that show popular classifiers like decision tree, random forest and SVM can also be used to generate rules [20, 2, 24, 19, 7, 3, 8]. We discuss the detail of the related work in Section 6.

Example [continued]: Table 2 presents a set of rules associated with the Samsung TL225 camera and tags in Table 1. Illustrating one of the rules: [Front LCD=1.5, Touchscreen=true, Gesture Control=true] $\rightarrow$ short battery life with $p = 0.13$ indicates that with probability of 0.13 the camera’s dual LCD feature along with its touchscreen and gesture control interfaces are responsible for the camera receiving the tag short battery life.

For an item $i$ having attributes values $\{a_{i1}, a_{i2}, \ldots, a_{im}\}$, if there are several rules for a tag $t_x$, the one with highest probability $p$ would be selected. For the rest of the paper, we use the example in Tables 1 and 2 as the running example.

In this paper, our objective is to identify the top-$k$ tags $T^* = \{t_1, t_2, \ldots, t_k\}$ for a user $u \in U$ and an item $i \in I$ such that $u$ can review $i$ by choosing from $T^*$. The result set $T^*$ is selected from the tag vocabulary $T$ if they are “meaningful”. Before formalizing the problem, let us define the essential characteristics that tags in $T^*$ must satisfy:

Relevance: Given item $i$ and tag vocabulary $T$, the relevance of a tag $t_x \in T^*$ denotes how well $t_x$ describes $i$. Mathematically, it is measured as the probability of obtaining $t_x$ given $i$, i.e., $REL(t_x,i) = Pr(t_x|i)$. As we have discussed earlier this score can be computed by employing a classifier modeling the relationship between item attributes and tags. Thus, $REL(T^*) = FUNC_{t_x \in T^*}(REL(t_x,i)) = \sum_{t_x \in T^*} FUNC_{t_x}(REL(t_x,i))$.

Given a list of tags $T$ which is sorted by the relevance (i.e., $REL(t_x,i) = Pr(t_x|i)$), the maximum relevance score is the total score for the top $k$ tags in the sorted list. We represent the maximum relevance score for a set of $k$ tags from $n$ tags in $T$ as $REL_{T^k}$.

Coverage: Given item $i$, tag vocabulary $T$, and a set of associated rules $R = \{\{a_i \rightarrow t_x\}\}$, the coverage of a tag $t_x \in T^*$ for the set of distinct item attribute values have been covered by it. We say $t_x$ covers the attribute value $a_{ij}$ if $a_{ij} \in \{a_i\}$, i.e., $COV(t_x, i) = \{a_i\}$. Therefore, $COV(T^*) = FUNC_{t_x \in T^*}(COV(t_x,i))$. We will discuss $FUNC$ in details later in Section 2.4.

Polarity: Given item $i$, and tag vocabulary $T$, the polarity of $T^*$ for a user reviewing item $i$ captures the distribution of sentiment in opinion. It is measured as the ratio of the number of the positive tags to the number of the negative tags, i.e., $POL(T^*) = \frac{|T^+|}{|T^-|}$.

While maximization of the first two characteristics, i.e., relevance and coverage, for determining the set $T^*$ of top-$k$ tags is obvious, the third characteristics, i.e., polarity is dependent on a user’s personal preference. Some users tend to be lenient and provide mostly positive feedback; some tend to be harsh. Thus, there is not any obvious way of estimating a user’s criticalness in reviewing. One reasonable solution is to aggregate sentiments of user demographic groups and consider the value of the group to which the user belongs as her reviewing tendency. For example, if the average rating for cameras by all young female users living in California is 8.0 (on a scale of 10.0), then a user belonging to the sub-population will have a criticalness factor of 0.8 (on a 0-1 scale); she is likely to assign 80% positive feedback and 20% negative feedback to a camera. $POL(T^*) = \frac{|T^+|}{|T^-|}$ should be at least $0.8/0.2 = 4$. In other words, polarity is the “odds” of the positive tags which is the probability of positive tags $\frac{|T^+|}{|T^-|}$ to the probability of negative tags $\frac{|T^-|}{|T^-|}$.

Since our TagAdvisor problem focuses on modeling the relationship between item attributes and tags, we leverage user personal preference as a parameter in our framework. We refer to this parameter, denoted by $\alpha$ as User Factor, where the value of the $\alpha$ is normalized to a [0,1] continuous sentiment scale.

2.2 The Problem

A user can review an item in different ways. A user can express her opinion on multiple item attributes which in turn, can either be positive or negative. For example, the set of tags $\{\text{great picture quality, great sharpness, great color accuracy, short battery life}\}$ contains positive feedback for the camera’s image quality and negative feedback for the camera’s battery life. Again, a user can express both positive and
negative opinion for the same attribute (or, set of attributes). For example, the set of tags \{short battery life, stylish\} contains both positive and negative feedback for the camera’s innovative/aspects (i.e., dual-screen, touchscreen and gesture-controlled). From Table 2, short battery life and stylish are tags related to camera attributes Front LCD, Touchscreen and Gesture Control for Samsung TL225.

We first propose a general TagAdvisor problem and then present two different problem instances that enable a wide range of real-world scenarios. The instances are distinct by the difference in formulation of the coverage of a set of tags $T^*$, i.e., $\text{cov}(T^*)$.

**Definition 1.** TagAdvisor Problem (TA): Given a set of rules $\mathcal{R} = \{(a,v) \rightarrow t_x\}$ for an item $i = \{a,v, a.v_2, \ldots\}$ and $t_x \in T$, non-negative integer budget $k$, relevance parameter $\beta$ ($0 \leq \beta \leq 1$), and user factor $a$ ($0 \leq a \leq 1$), find a subset of $T^* \subseteq T$ such that:

- $|T^*| \leq k$;
- $\text{pol}(T^*) \geq \frac{a}{1-a}$;
- $\text{rel}(T^*) \geq \beta \times \text{rel}_{\text{max}}(T^*)$;
- $\text{cov}(T^*)$ is maximized,

where $\text{rel}(T^*)$ is the total relevance of tags in $T^*$, $\text{rel}_{\text{max}}(T^*)$ is the maximum relevance for $k$ tags from $T$, $\text{cov}(T^*)$ is the total number of item attribute values covered by tags in $T^*$, and $\text{pol}(T^*)$ is the sentiment in opinion by tags in $T^*$. The relevance parameter $\beta$ ensures that the relevance score of tags in $T^*$ is as close to the best possible relevance score $\text{rel}_{\text{max}}(T^*)$. The user factor $a$ denotes the proportion of positive and negative tags preferred by a user.

### 2.3 General Model

We model the TagAdvisor Problem as bipartite graph $G_{TA} = (V = V_T \cup V_I, E)$ as shown in Figure 1, where $V_T$ is the set of nodes associated with the tag vocabulary $T$, $V_I$ is the set of nodes associated with the item attribute values, $V_T$ and $V_I$ are disjoint and $E \subseteq (V_T \times V_I)$. The nodes in partite $V_T$ are further classified into positive nodes $V_T^+$ (colored green) and negative nodes $V_T^-$ (colored red), based on the sentiment of the tags. If the same tag has positive sentiment for an attribute value and negative sentiment for another attribute value, we consider the tag as two different nodes in the set $V_T$. An edge $(t_x, a.v_y) \in E$ if $t_x$ covers attribute value $a.v_y$, i.e., the rule $\{(a,v) \rightarrow t_x\}$, $a.v_y \in \{a,v\}$ exists; similarly $(t_x, a.v_y) \in E$ if $t_x$ covers attribute value $a.v_y$. We use the graph model for the coverage purpose.

**Example [continued]:** Figure 2 shows the bipartite graph model of our running example in Table 2, where $G_{TA}$ has two parts $V_T = V_T^+ \cup V_T^-$ in green and red respectively and $V_I$ in yellow, where $T^+ = \{t_1^+, t_2^+, t_3^+, t_4^+, t_5^+, t_6^+\}$, $T^- = \{t_1^-, t_5^-, t_6^-\}$. The edges represent the rules in Table 2. For example, nodes $t_1^+$ and $t_3^+$ have three edges to the same attribute value nodes $a.v_2$, $a.v_7$, and $a.v_8$.

We next define two concrete problem instances of the TA Problem based on $\text{cov}(T^*)$.

### 2.4 Concrete Problem Instances

In the first problem, $\text{cov}(T^*)$ is defined as the total number of item attribute values covered by the tags in $T^*$, independent of their sentiment. In this problem, an attribute value $a.v_y$ for an attribute $a_y$ of an item $i$ is covered by $T^*$ if $\exists t_x \in T^*$ such that $a.v_y \in \text{cov}(t_x, i)$, i.e., there exists a tag $t_x$ covering $a.v_y$, independent of its sentiment.

**Definition 2.** Given a set of tags $T^*$, INDEPENDENT-COVERAGE of $T^*$ is defined as:

$$\text{cov}_{IC}(T^*) = \sum_{t_x \in T^*} |\text{cov}(t_x, i)|$$

**Example [continued]:** In the running example in Table 2 and by Figure 2, if $T^* = \{t_1^+, t_2^+, t_6^+\} = \{t_1, t_2, t_6\} = \{\text{super cool}, \}$
Figure 1: TagAdvisor Bipartite Graph model.

Figure 2: TagAdvisor Bipartite Graph model of the Running Example.

stylish, gimmicky touchscreen], then $\text{COV}_{IC}(T^*) = |\{a_3, a_4, a_5, a_7, a_8\}| = |\{\text{Color} = \text{Red}, \text{Front LCD} = 1.5", \text{Back LCD} = 3.5", \text{Touchscreen} = \text{true}, \text{Gesture Control} = \text{true}\}| = 5$.

Based on $\text{COV}_{IC}(T^*)$ in Equation 1 the first problem can now be defined as follows.

**PROBLEM 1. [Independent-Coverage TA Problem (ICA-TA)]:** This problem is an instance of TagAdvisor Problem (TA) in Definition 1. where the input and constrains are the same but the objective is:

* $\text{COV}_{IC}(T^*)$ (given by Equation 1) is maximized

However, by considering the coverage of an item attribute value by a tag independent of the tag’s sentiment, we may restrict a user from reviewing both positively and negatively about the different aspects of an item. By $\text{COV}_{IC}(T^*)$, if $T^*$ includes a tag that is positive and covers a subset of item attribute values, another tag that is negative and covers the same subset would not be included in $T^*$. In the running example in Table 2, if at least one of the positive tags, say $t^+_1$ : stylish belongs to $T^*$ with a higher relevance score, then $a.v_7$: Touchscreen = true and $a.v_8$: Gesture Control = true are considered covered because of rule $p$: $\{a.v_3, a.v_4, a.v_7, a.v_8\} \rightarrow t^+_1$; $T^*$ would not include either of the negative tags gimmicky touchscreen and poor battery life related to $a.v_7$ and $a.v_8$. This motivates us to define the second problem instance where an item attribute value is considered fully covered if it is covered by both positive and negative tags.

**DEFINITION 3.** Given a set of tags $T^*$, **Dependent-Coverage of $T^*$** is defined as:

$$\text{COV}_{DC}(T^*) = \left| \bigcup_{t^+_i \in T^*} \text{COV}(t^+_i, i) \right| \cap \left| \bigcup_{t^-_i \in T^*} \text{COV}(t^-_i, i) \right| + \left| \bigcup_{t^+_i \in T^*} \text{COV}(t^+_i, i) \right| \cap \left| \bigcup_{t^-_i \in T^*} \text{COV}(t^-_i, i) \right| + \left| \bigcup_{t^-_i \in T^*} \text{COV}(t^-_i, i) \right| \cap \left| \bigcup_{t^+_i \in T^*} \text{COV}(t^+_i, i) \right|$$

(2)

In second problem, coverage of $a.v_y$ depends on the sentiment of its associated tags. An attribute value $a.v_y$ for an attribute $a_y$ of an item $i$ is covered if one of the following holds:

* $a.v_y$ is covered by both positive and negative tags, and atleast one of its positive and atleast one of its negative tags belong to $T^*$. Formally, $\exists t^+_i \in T^*$, $\exists t^-_i \in T^*$ such that $a.v_y \notin \text{COV}(t^+_i, i) \cap a.v_y \in \text{COV}(t^-_i, i)$

* $a.v_y$ is covered only by positive tags and not negative tags, and atleast one of its positive tags belongs to $T^*$. Formally, $\exists t^+_i \in T^*$, $\forall t^-_i \in T^*$ such that $a.v_y \notin \text{COV}(t^+_i, i) \cap a.v_y \notin \text{COV}(t^-_i, i)$

Thus the coverage function in this problem variant considers both positive and negative tags for an attribute value if it exists; otherwise, it focuses on either the positive tag or the negative tag (which ever exists) and ends up returning the same $T^*$ as Problem 1. In our running example in Table 2, we see that attribute $a.v_4$: FrontLCD = 1.5” is in three rules corresponding to tags [super cool, stylish, and poor battery life]. By this definition of coverage, $a.v_4$: FrontLCD = 1.5” is covered by a tag in $T^*$ if at least one of the positive tags [super cool or stylish] and the one negative tag poor battery life exists in $T^*$. Again, $a.v_3$: Color = Red is covered if the positive tag stylish belongs to $T^*$ since there is no negative tag related to $a.v_3$ in the rules in Table 2 and $a.v_6$: ShutterSpeed = 8 – 1/2000 is covered if blurry pictures is in $T^*$ since there is no positive tag related to $a.v_6$ in the rules in Table 2.

**EXAMPLE [continued]:** In the running example in Table 2 and by Figure 2, if $T^* = \{t^+_1, t^+_2, t^-_3\} = \{\text{super cool, stylish, gimmicky touchscreen}\}$, then $\text{COV}_{DC}(T^*) = 4$:

* $a.v_4$ is covered only by negative tags and not positive tags, and atleast one of its negative tags belongs to $T^*$. Formally, $\exists t^-_i \in T^*, \forall t^+_j \in T^*$ such that $a.v_y \notin \text{COV}(t^-_i, i) \cap a.v_y \notin \text{COV}(t^+_j, i)$

Thus the second problem can now be defined as follows.

**PROBLEM 2. [Dependent-Coverage TA Problem (DC-TA)]:** This problem is an instance of TagAdvisor Problem (TA) in Definition 1. where the input and constrains are the same but the objective is:

* $\text{COV}_{DC}(T^*)$ (given by Equation 2) is maximized

3. **Independent-Coverage TagAdvisor (IC-TA)**

In this section, we first analyze the computational complexity of the Independent-Coverage TagAdvisor (IC-TA) problem and show that it is NP-complete; then we discuss an exact algorithm and an approximation algorithm for solving it.

3.1 **Computational Complexity**

The decision version of the IC-TA is defined as follows:

Given a set of rules $\mathcal{R} = \{\{a.v\} \rightarrow t_i\}$ for an item $i$, non-negative integer budget $k$, relevance parameter $\beta (0 \leq \beta \leq 1)$, user factor $\alpha (0 \leq \alpha \leq 1)$, and integer threshold $\gamma \geq 0$, is there
a set of $T^* \subseteq T$ such that $\text{COV}_{IC}(T^*) \geq \gamma$ subject to: $|T^*| \leq k$, $\text{POL}(T^*) \geq \frac{\alpha}{|x|}$, and $\text{REL}(T^*) \geq \beta \cdot \text{REL}_{\max}$. 

**Theorem 1.** The decision version of the Independent-Coverage TagAdvisor (IC-TA) problem is NP-Complete.

**Proof.** The membership of decision version of IC-TA in NP is obvious. To verify NP-Completeness, we reduce Max-Coverage problem with group budget constraints (MCG) [5], to our problem and argue that a solution to (MCG) exists, if and only if, a solution to our problem exists. In MCG problem, given $S = \{S_1, S_2, \ldots\}$ as a collection of sets where each set $S_i$ is a subset of a ground set $X$ of elements and $S$ is partitioned into groups $G_1, G_2, \ldots, G_m$, the goal is to pick $k$ sets from $S$ such that at most $k_i$ sets be picked from each group $G_i$ and cardinality of their union is maximum. This problem was proved to be NP-Complete by reduction from Max-Coverage in [5] if the number of groups is at least one, i.e., $m \geq 1$. We construct an instance of IC-TA problem such that the solution for MCG with two groups $m = 2$ exists, and if only if, the solution to our IC-TA instance exists.

We create a set of rules $\mathcal{R} = \{a.e \rightarrow t_x\}$ such that for every element in ground set $X$, there is a corresponding $a.e \in \{a.e\}$ and for every $S_i \in S$, there exists a corresponding $t_x \in \mathcal{R}$. Next, based on the sentiment of the tags, we partition $\mathcal{R}$ into two groups, i.e., positive and negative groups where $G_1$ corresponds to positive group and $G_2$ corresponds to negative group. We set the $\alpha = \frac{k_1}{k_2}$, where $k_1$ is the number of sets that should be accessed from each group $G_1$, and $\beta = 0$ i.e., the polarity constraint $\text{POL}(T^*) \geq \frac{\alpha}{|x|}$ is satisfied and relevance constraint will be relaxed because $\text{REL}(T^*) \geq 0$ is always true. In Equation 1, $\text{COV}_{IC}(T^*)$ is the cardinality of the union of the coverage of the tags. Thus, in this IC-TA instance, if $T^*$ with $k = k_1 + k_2$ tags, where $k_1$ tags are accessed from positive group and $k_2$ tags are selected from negative group maximizes the $\text{COV}_{IC}(T^*)$, then the corresponding sets in $S$ maximizes the cardinality of their union in MCG with two groups. Thus, IC-TA problem is NP-Complete. 

### 3.2 Exact Algorithm (E-IC-TA)

A brute-force approach to solve the IC-TA problem enumerates all possible $\binom{|x|}{k}$ (n is the total number of tags in vocabulary, $k$ is the size of $T^*$) combinations of tags in order to return the optimal set of tags maximizing coverage $\text{COV}_{IC}(T^*)$ and satisfying the constraints. The number of possible candidate sets is exponential in the number of the rules for an item. If there are $m$ boolean attributes for an item, there are potentially $2^m$ rules for tags. Thus, evaluating the constraints on each of the candidate sets and selecting the optimal result can be prohibitively expensive. Although general purpose pruning-based optimization techniques (such as branch and bound algorithms) can be used to solve the problem more efficiently, they are only limited to finding the top-1, and it is not clear how to extend them for top-$k$, $k > 1$. We refer to this naive exact algorithm of IC-TA as E-IC-TA and develop a practical and efficient algorithm to solve it.

### 3.3 Approximation Algorithm (A-IC-TA)

In order to solve IC-TA problem, we consider the Max-Coverage problem with group budget constraints (MCG) problem variant in Chekuri et al.'s paper [5], where given $S = \{S_1, S_2, \ldots\}$ as a collection of sets where each set $S_i$ is a subset of a ground set $X$ and $S$ is partitioned into groups $G_1, G_2, \ldots, G_m$, the goal is to pick $k$ sets from $S$ such that at most $k_i$ be picked from each group $G_i$ and cardinality of their union is maximum. The authors in [5] proposed a greedy solution with a 2-approximation algorithm.

In our problem, the set $S$ is the set of rules $\mathcal{R} = \{\{a.e\} \rightarrow t_x\}$ which is partitioned into two groups based on the tags sentiments.

**Algorithm 1: IC-TA Algorithm (A-IC-TA)**

**Input:** Tag vocabulary $T$, set of rules $\mathcal{R} = \{a.e \rightarrow t_x\}$, budget $k > 0$, relevance parameter $0 < \beta \leq 1$, user factor $0 < \alpha \leq 1$

**Output:** set of tags $T^* \subseteq T$ of size $k$

1. $k_1 = \lceil k \cdot \alpha \rceil$, $k_2 = k - k_1$;
2. $T^* = \emptyset$;
3. for $x \gets 0$ to $k$ do

   4. for $t_y \in T \setminus T^*$ do

      5. if $t_y \in T^*$ and $|T^*| < k_1$ then

         maxRel = $\text{argmax}_{t_x \in T^{+*}} \text{REL}(t_x)$;

         $t_x \in T^{+*}$

      6. if $\text{REL}(t_x) \geq \beta \cdot \maxRel$ then

         Compute $\text{COV}_{IC}(T^* \cup t_y)$;

         $t_y \in T^*$

      7. if $t_y \in T^*$ and $|T^*| < k_2$ then

         maxRel = $\text{argmax}_{t_x \in T^{-*}} \text{REL}(t_x)$;

         $t_x \in T^{-*}$

      8. if $\text{REL}(t_x) \geq \beta \cdot \maxRel$ then

         Compute $\text{COV}_{IC}(T^* \cup t_y)$;

      10. $t_y \in T^*$

9. return $T^*$

We use the similar greedy approach in [5] and check an extra constraint for the relevance. Intuitively, the greedy approach will iteratively picks those relevant tags that cover the maximum number of uncovered item attribute values.

Algorithm 1 is the pseudo code for our algorithm, denoted as A-IC-TA. The A-IC-TA algorithm iteratively picks tags from $T$ that cover the maximum number of uncovered item attribute values such that the number of positive and negative tags are $k_1 = \lceil k \cdot \alpha \rceil$, $k_2 = k - k_1$ and $\text{REL}(T^*) \geq \beta \cdot \text{REL}_{\max}$. If we assume all tags in $T$ are sorted by their relevance, the $\text{REL}_{\max}$ is the summation of the first $k$ tags in the sorted list. More specifically, let us assume a positive tag $t_y$ is picked and let $t_x \in T^* \setminus T^*$ be the positive tag with highest relevance score (maxRel) which have not been selected yet (line 6). Algorithm 1 iteratively adding one tag with highest coverage to $T^*$, where its relevance score is at least $\beta \cdot \maxRel$ (lines 4 – 12).

**Theorem 2.** The A-IC-TA Algorithm provides near optimal solution with 2-approximation factor.

**Proof.** The proof follows from the 2-approximation factor proof of the algorithm for solving the Max-Coverage with group budget constraints (MCG) problem in [5].

**Example** [continued]: In the running example, for $k = 2$, $\alpha = 0.5$, and $\beta = 0.5$, Algorithm 1 returns $T^*$ = [stylish, blurry pictures]. In first iteration, the highest relevance score of the positive tags (maxRel) is 0.3. Among the positive tags super cool and stylish has relevance larger than 0.15 = 0.5 - 0.3 and coverage score 3 and 4. Thus stylish with highest coverage score of 4 will be selected. Next, maxRel will be updated to 0.15 which is the highest relevance score of the negative tags then, among all the negative tags whose relevance are larger than 0.075 = 0.5 - 0.15, blurry pictures with highest coverage of 3 will be selected.

### 4. Dependent-Coverage TagAdvisor (DC-TA)

In this section, we focus on the Dependent-Coverage TagAdvisor (DC-TA) problem. We first propose a graph model for the problem,
then analyze its computational complexity and prove that it is NP-complete, and finally develop an exact algorithm and an efficient constant factor approximation algorithm for solving it.

In order to solve the DC-TA problem, we transform the bipartite graph in Figure 1 to a complete weighted graph \( G_{DC-TA} = (V_T, E) \), where \( V_T \) is the set of nodes associated with the tag vocabulary \( T \), and \( E \subseteq (V_T \times V_T) \). Each edge \( e \in E \) has a weight, \( w : E \rightarrow \mathbb{R} \). Let us define the edge weight as the distance (or, dissimilarity) between two tag nodes, i.e., \( w(v_{t_1}, v_{t_2}) \) where \( v_{t_1}, v_{t_2} \in V_T \). We can consider each tag as a boolean vector of size \( m \) (number of item attributes) where bit at location \( y \) is 1 if \( a_{t,y} \in COV(t_x, t_{\ast}) \). Using such a vector representation of the tags, we used Hamming metric to measure the distance \( w(v_{t_1}, v_{t_2}) \). In our framework, \( T \) is partitioned into two disjoint sets: \( T^+ \) and \( T^- \) based on tag sentiment. Thus, there can be three kind of node-to-node connectivity: \( v_{t_1}^+ \) (\( t_1 \in T^+ \)) is connected to \( v_{t_2}^+ \) (\( t_2 \in T^+ \)), \( v_{t_1}^- \) (\( t_1 \in T^- \)) is connected to \( v_{t_2}^- \) (\( t_2 \in T^- \)), and \( v_{t_1}^- \) (\( t_1 \in T^- \)) is connected to \( v_{t_2}^+ \) (\( t_2 \in T^+ \)). The first two connectivities are intra-edges and the third belongs to the category of cross-edges.

Recall that the the coverage function \( COV_{DC}(T^*) \) discussed in Equation 2 is based on three conditions that considers both positive and negative tags for an attribute value if it exists; otherwise, it focuses on either the positive tag or the negative tag. We argue that we can reduce the last two conditions to the first one by introducing dummy edges. In other words, for attribute values with only positive tags, selecting any negative tag would not influence their coverage; hence we can add dummy edges from those attribute value nodes to all the negative tags. Similarly for attribute values with only negative tags, selecting any positive tag would not influence their coverage and we can add dummy edges from those attribute value nodes to all the positive tags.

Figure 3 shows the original bipartite graph in Figure 1 with dummy edges for the running example in Table 2. Since node Colors=Red is not covered by any of the negative tag nodes [poor battery life, blurry pictures, gimmicky touchscreen], we add dummy edges (red dotted lines) from it to all the negative tags. Similarly, the dummy edges (green dotted lines) are added from Shutter Speed=8-1/2000 to all positive tag nodes [super cool, stylish, lightweight]. Figure 4 shows the graph \( G_{DC-TA} \) of our running example in Table 2 having 6 nodes \( T = \{t_1^+, t_2^+, t_3^+, t_1^-, t_2^-, t_3^-\} = [\text{super cool, stylish, lightweight, poor battery life, blurry pictures, gimmicky touchscreen}] \); the edge weights show the dissimilarity between two tags measured by Hamming metric. For example, the distance between the \( t_1^+, t_2^- \) is calculated as 1 because \( t_1^+, t_2^- \) is connected to \( a_{n LCD=1.5}, a_{n Shutter Speed=8-1/2000}, a_{n Touchscreen=true}, a_{n Gesture Control=true} \). The vector representation of \( t_1^+ \) is \( [0, 0, 0, 1, 0, 1, 1] \). Similarly, \( t_2^- \) can be represented as \( [0, 0, 1, 1, 0, 0, 1, 1] \) and their Hamming distance is 2.

Our objective in this problem is to maximize \( COV_{DC}(T^*) \). Considering this transformed complete weighted graph model, we would select positive tags \( T^* \) and negative tags \( T^- \) from nodes in \( V_{T^+} \) and \( V_{T^-} \) respectively such that the constraints are satisfied and the total weight of the cross-edges is minimum in the induced graph. We should not select nodes in \( V_{T^+} \) (or, \( V_{T^-} \)) which cover similar attribute values. To ensure diversity, we should also maximize the total weight of the intra-edges in each of the positive node set and negative node set in the induced graph. Formally, the objective of DC-TA in this graph model is to minimize:

\[
\vartheta_{DC}(T^*) = \sum_{t_e \in T^+} w(v_{t_e}, v_{t_e}) - \sum_{t_e, t_e \in T^+} w(v_{t_e}, v_{t_e}) \tag{3}
\]

Where the first term is summation of the weights of the cross-edges (edges between positive-negative tags) and the second term is the summation of the weights of the intra-edges (edges between positive-positive and negative-negative tags). We can observe that minimizing \( \vartheta_{DC}(T^*) \) is equivalent to maximizing the \( COV_{DC}(T^*) \). \( COV_{DC}(T^*) \) is based on the three different conditions over the item attribute values. Due to the inclusion of dummy edges, the problem reduces to one condition which is maximizing the similarity of positive and negative tags. Clearly, minimizing the positive and negative tags dissimilarity by Equation 3 is equivalent to maximizing the similarity of those tags. Next we analyze the computational complexity of this problem.

### 4.1 Computational Complexity

The decision version of the DC-TA is defined as follows:

Given graph \( G_{DC-TA} = (V_T, E) \), non-negative integer budget \( k \), relevance parameter \( \beta (0 \leq \beta \leq 1) \), user factor \( \alpha (0 \leq \alpha \leq 1) \), and integer threshold \( \gamma \geq 0 \), is there a set of \( T^* \subseteq T \) such that \( \vartheta_{DC} \leq \gamma \) subject to: \( |T^*| \leq k \), \( POL(T^*) = \frac{\alpha}{1-\alpha} \), and \( REL(T^*) \geq \beta \cdot REL_{max} \).

**Theorem 3.** The decision version of the Dependent-Coverage TagAdvisor (DC-TA) problem is NP-Complete.

**Proof.** It is obvious that the decision version of the DC-TA is in NP. To verify NP-Completeness, we reduce the MAX-SUM Facility Dispersion problem [22, 9, 12] to our problem and argue that a solution to MAX-SUM Facility Dispersion exists, if and only if, a solution to our problem exists. In MAX-SUM Facility Dispersion problem, given a set of \( V = \{v_1, v_2, ..., v_n\} \) of \( n \) nodes, a non-negative distance \( w(v_i, v_j) \) for each pair of nodes \( v_i, v_j \), and an integer \( k \) smaller than \( n \), the goal is to find a subset \( P = \{v_1, v_2, ..., v_p\} \) of \( V \), with \( |P| = p \), such that sum of distances are maximized. This problem was proved to be NP-Complete even when the edge weights satisfy the triangle inequality [9, 12]. We construct an instance of DC-TA problem such that the solution for MAX-SUM Facility Dispersion exists, if and only if, the solution to our DC-TA instance exists.

We create a graph \( G_{DC-TA} = (V_T, E) \) such that for every \( v_i \in V \) there is a corresponding node \( v_{t_i} \in V_T \) and a distance \( w(v_i, v_{t_j}) \) corresponds to the Hamming distance of two tags \( t_{x_1} \) and \( t_{x_2} \), i.e., \( w(v_{t_1}, v_{t_2}) \). Let in this DC-TA instance, \( \alpha = 1 \), \( i_1 = 1 \), \( k_1 = 1 \), and \( k_2 = 0 \) (only \( p \) positive tags should be selected). Also by setting \( \beta = 0 \) the relevance constraint will be relaxed because \( REL(T^*) \geq 0 \) is always true. Let in DC-TA instance, positive and negative tags cover exactly same item attribute values, i.e., the distance between positive and negative tags be 0. Thus, the DC-TA problem collapses to that of finding \( p \) positive tags such that \(- \sum_{t_e \in T^+} w(v_{t_e}, v_{t_e}) \) is minimum or the sum of distances are maximum and \( COV_{DC}(T^*) \) is maximized. Thus, in this DC-TA instance, if \( T^p \) with \( p \) positive tags and zero negative tags maximizes the \( COV_{DC}(T^*) \), then the corresponding nodes in \( V \) maximizes the sum of distances in MAX-SUM Facility Dispersion. Thus, DC-TA problem is NP-Complete.

### 4.2 Exact Algorithm (E-DC-TA)

Similar to Section 3.2, a brute-force approach to solve the DC-TA problem enumerates all possible \( C_k \) combinations of tags in order to return the optimal set maximizing coverage \( COV_{DC}(T^*) \) (or, minimizing \( \vartheta_{DC}(T^*) \)) and satisfying the constraints. We refer
to this computationally prohibitive exact algorithm of DC-TA as E-DC-TA and develop an efficient algorithm for this problem.

4.3 Approximation Algorithm (A-DC-TA)

Given graph \(G_{DC-TA} = (V_T, E)\) as DC-TA model, relevance parameter \(\beta\), and user factor \(\alpha\), the goal is to select \(k_1 = \lfloor \alpha k \rfloor\) positive tags and \(k_2 = k - k_1\) negative tags such that \(\text{REL}(T^+) \geq \beta\) and \(\vartheta_{DC}(T^+)\) is minimum.

First, we show that such \(\vartheta_{DC}(T^+)\) is not submodular. In submodular functions the incremental gain of adding an element to a set decreases as the size of the set increases, i.e., in the context of our paper, for all tags \(t_x\) and \(S \subseteq T^+\), \(F(S \cup \{t_x\}) - F(S) \geq F(T^+ \cup \{t_x\}) - F(T)\). The authors in [18] proved that if a function is monotone and submodular, the greedy approach provides near optimal solution with \((1 - 1/e)\)-approximation factor. We prove that \(\vartheta_{DC}(T^+)\) is not submodular, thus, there is no any greedy approach provides near optimal solution with \((1 - 1/e)\)-approximation factor for DC-TA problem. Next we propose an approximation algorithm, denoted by A-DC-TA and we prove its approximation factor.

**Theorem 4.** The function \(\vartheta_{DC}(T^+)\) is not submodular.

**Proof.** Let \(T_1 = T^+_1 \cup T_1^-\) be the set of positive and negative tags for item \(i\) covering item attribute values \(\{a.v\}\) with \(I_{A1} \subseteq I_A\) as schema. Let \(T_2 \subseteq T_1\) covers attribute values \(\{a.v\}\) with schema \(I_{A2} \subseteq I_{A1}\), such that \(T_2\) has the same positive tags \(T_2^+ = T_1^+\) but \(T_2^-\) has more negative tags than \(T_2^-\) i.e., in \(T_1\) there are some values for attributes \(\{a\}\) that are cover by negative tags, \(\{a\} \subseteq I_{A1}\), which those attribute values are not cover by \(T_2\), \(\{a\} \not\subseteq I_{A2}\). Now assume we want to add to both sets a positive tag \(t_{x_2}\) that covers some values of attributes \(\{a'\} \subseteq \{a\}\). In DC-TA problem every attribute values associated with both positive and negative tags is covered if at least one from each negative and positive tags are selected. It is clear that adding \(t_{x_2}\) to \(T_1\) is more beneficial than adding it to \(T_2\) because all values of attributes \(a_j \in \{a'\}\) are covered by \(T_1\) by both positive and negative tags but they are only covered by \(T_2\) by positive tag but not negative. Thus, the incremental gain of adding this tag to a set increased as the size of the set increases, which contradicts with submodularity, where the incremental gain of adding a tag to a set should decreases as the size of the set increases.

In DC-TA, the goal is to select \(k_1 = \lfloor k \alpha \rfloor\) and \(k_2 = k - k_1\) tags from the \(T^+\) and \(T^-\) such that the induced subgraphs of intradeces with \(k_1\) and \(k_2\) tags in \(T^+\) and \(T^-\) are actually the maximum cliques where the induced bichlique of the cross edges between \(k_1\) and \(k_2\) tags has minimum bichliqye. To the best of our knowledge this is the first time that the problem with the combination of the maximum cliques and minimum bichlique is defined. We propose a greedy algorithm and theoretically prove that it produces a solution with constant factor approximation of the optimal.

The A-DC-TA Algorithm 2 uses the user factor \(\alpha\) to find the number of positive and negative tags need to be selected from each partition, i.e. \(k_1\) and \(k_2\). Let \(t_x \in T^+ \setminus T^+\) and \(t_y \in T^- \setminus T^-\) be the tags with highest relevance score in positive and negative tags which have not been selected yet. The \(\text{maxRel}\) in line 4, is the summation of the relevance score of \(t_x\) and \(t_y\). Lines 3 – 5 of the algorithm iteratively picks the minimum weight cross-edges \((v_{tx}, v_{ty}), t_x \in T^+\), \(t_y \in T^-\) with the relevance score of at least \(\beta\)-\(\text{maxRel}\) and adds those tags to the \(T^+\) until the number of selected positive or negative tags be \(k_1\) or \(k_2\). If the number of selected positive and negative tags is \(k_1\) and \(k_2\), the algorithm returns \(T^+\) as the top-\(k\) tags, otherwise there are still more tags that should be selected from either positive or negative tags (not both). Let us assume \(k_1\) positive tags are selected. The algorithm (line 9 – 10) finds the \(t_x, t_y \in T^- \setminus T^-\) which are the tags with highest relevance score in negative tags which have not been selected yet. The \(\text{maxRel}\) in line 10, is the summation of their relevance scores. Then, it selects the maximum intra-edges \((v_{tx}, v_{ty}), t_x \in T^+, t_y \in T^-\) with the relevance score of at least \(\beta\)-\(\text{maxRel}\).

**Example [continued]:** In the running example, for \(k = 2, \alpha = 0.5, \beta = 0.5\), solving the problem with practical heuristic Algorithm 2 returns \(T^+ = \{\text{stylish}, \text{poor battery life}\}\). It first finds \(t_{x_2} = \text{super cool}\) and \(t_{y_2} = \text{gimmicky touchscreen}\) as the positive and negative tags with highest relevance scores 0.3 and 0.15 (\(\text{maxRel} = 0.45\)). Then it selects \(t_{x_2} = \text{stylish}\) and \(t_{y_2} = \text{poor battery life}\) because among the cross-edges it has the lowest weight 1. Then it checks the relevance constraint, i.e. \(\text{REL}(\{t_x, t_y\}) > \beta\)-\(\text{maxRel}\). Since their relevance 0.33 = 0.2 + 0.13 is greater than 0.225 = 0.5 · 0.45, they added to \(T^+\).

**Theorem 5.** The proposed heuristic DC-TA algorithm 2 produce a solution with 2-approximation of the optimal, i.e. \(\vartheta_{DC}(T^\alpha) \leq 2 \cdot \vartheta_{DC}(T^\alpha)\).

**Proof.** Please refer to Appendix 10.1.

5. EXPERIMENTS

5.1 Experimental setup

**System configuration:** Our prototype system is implemented in Java with JDK 5.0. All experiments were conducted on an Ubuntu machine with 2.0GHz Intel processor and 8GB RAM. All numbers are obtained as the average over 10000 runs.

**Datasets:** We conduct a comprehensive set of experiments using both synthetic and real data crawled from the web to evaluate efficiency and quality of our proposed algorithms. For synthetic data,
we generated a large boolean matrix of item attributes with positive and negative tags. For real data, we crawled Yahoo! Autos, Walmart and Google Product for building a car dataset and a camera dataset. We use the synthetic dataset for quantitative experiments, and the real dataset for qualitative study. The details of each dataset is described below:

**Synthetic Dataset:** We generate a large boolean matrix of dimension 10,000 (items)×100 (50 attributes + 25 positive tags + 25 negative tags). We split the 50 independent and identically distributed attributes into four groups, where the value is set to 1 with probabilities of 0.75, 0.15, 0.10 and 0.05 respectively. For each of the 50 tags, we randomly picked a set of attributes that are correlated to it. A tag is set to 1 if majority of the attributes in its correlated set of attributes have boolean value 1.

**Real Camera Dataset:** We crawl a real dataset of over hundred cameras listed at Walmart 1. The Walmart camera data consists of 12,600 reviews from 11,500 users on 140 cameras. Since the camera information crawled from Walmart lacked well-defined item attribute values for all the cameras, we look up Google Products 2 and parse a total of 120 attributes such as self-timer, red-eye fix, auto focus, built-in flash, etc. We process the reviews to identify a set of positive and negative tags such as stunning photo quality, great pocket camera, short battery life, expensive, etc. using the keyword extraction toolkitAlchemyAPI 3 which, in turn, uses natural language processing technology and machine learning algorithms to extract semantic meta-data from content. We employ RIPPER 6 to predict the set of rules that shows the dependency between item attributes and tags.

**Real Car Dataset:** We crawl a real dataset of 100 used cars listed at Yahoo! Autos 4 for the year 2010. The products contain technical specifications as well as ratings and reviews, which include pros and cons. We parse a total of 47 attributes: 15 numeric, and 32 boolean and categorial (the latter is generalized to boolean). The total number of reviews, i.e., pros and cons by users for the 100 cars is 2350. Since a feedback is labelled ‘pro’ or ‘con’, we do not need to employ any external text mining toolkit for getting the sentiments. The feedbacks are short phrases and keywords. These phrases are processed by domain experts to identify 20 representative positive and 20 representative negative tags that cover all the keywords crawled. For example, the ‘pro’ keywords driver seat comfort, cockpit comfort including ability to reach all controls easily, comfort is truly exceptional, super comfy and roomy for 4 people and dog correspond to the representative positive tag comfortable.

### Performance Measures

Our quantitative performance indicators are (i) efficiency of the algorithms, (ii) approximation factor of results produced by the approximation algorithms, and (iii) quality of the results produced. The efficiency of our algorithms is measured by the overall execution time, whereas approximation factor is determined by the ratio of the approximate result score to the actual optimal result score. The quality of result is measured by the ratio of features covered by our algorithms to the total number of features. We show that our algorithms are scalable and achieve much better response time than the exact algorithm while maintaining similar result quality. In order to demonstrate that the top-k tags returned by our approaches are useful to the end users, we conduct a user study through Amazon Mechanical Turk as well as write interesting case study.

### 5.2 Experimental Results

#### 5.2.1 Quantitative Evaluation

We first compare the execution time of our approximation algorithms against the brute-force counterparts. Figure 5 shows that the execution time of the proposed algorithms A-IC-TA and A-DC-TA are several orders of magnitude faster than the corresponding exact algorithms E-IC-TA, and E-DC-TA for k=10, user factor $\alpha=0.5$, and relevance parameter $\beta=0.5$ on entire synthetic data. Figures 6 and 7 compare execution time of A-IC-TA with E-IC-TA and that of A-DC-TA with E-DC-TA respectively by varying parameter $k$, with $\alpha=0.5$, and $\beta=0.5$. We observe that by increasing $k$, execution time of the exact algorithm grows exponentially, while A-IC-TA and A-DC-TA scales well.

Next, we investigate the ratio of the approximate result score to the actual optimal result score. In A-IC-TA and A-DC-TA the approximation ratio is the value of the $\text{CON}_{IC}(T^*)$ and $\delta_{DC}(T^*)$ in Equation 2 to the optimal solutions. We proved in theorems 2 and 5, A-IC-TA and A-DC-TA produce solutions with 2-approximation of the optimal. Figure 8 shows that by varying $k$, the approximation ratios are less than 2.

Finally, we evaluate the quality of results returned by our approximation algorithms by measuring the proportion of tags covered by the result set of $k$ tags in $T^*$. We compare the proposed algorithms A-IC-TA and A-DC-TA with the exact algorithms E-IC-TA and E-DC-TA by using the Independent-Coverage function, $\text{CON}_{IC}(T^*)$, in Equation 1 and Dependent-Coverage function, $\text{CON}_{DC}(T^*)$, in Equation 2 respectively. We conduct our experiments with different set of constraint conditions, i.e., user factor ($\alpha$), relevance parameter ($\beta$), and $k$. First, we set $\alpha = 0.5$, $\beta = 1.0$, and vary $k$ from

---

1. www.walmart.com
2. www.google.com/about/products
3. www.alchemyapi.com
4. autos.yahoo.com
2 to 10 in Figure 9. The results show that by increasing number of tags $k$, the proportion of covered item attribute values are increased. Moreover, the quality of our A-IC-TA and A-DC-TA algorithms are almost as exact algorithms E-IC-TA and E-DC-TA. Second, we set $k = 10$, $\alpha = 0.5$, and relevance parameter $\beta$ varies from 0.1 to 0.9 in step of 0.2. Th results in Figure 10 shows that although the relevance is increasing, proposed A-IC-TA and A-DC-TA algorithms are able to find 10 tags with as high quality as the exact algorithms. Third, we set $k = 10$, $\beta = 0.5$, and user factor $\alpha$ varies from 0.1 to 0.9 in step of 0.2. Results are shown in Figure 11. As one can see from the figure, by increasing the user factor parameter the proportion of covered item attribute values is decreasing. In other words, there are some item attribute values that will be covered by negative tags and since the user factor is high, the lower negative tags are appeared which lead to lower quality. However, the results show that the quality of our algorithm is still as good as the exact algorithms. In summy, all the results from different set of constraint conditions confirm the fact that despite the significant reduction in execution time, our A-IC-TA and A-DC-TA algorithms do not compromise much in terms of analysis quality.

5.2.2 Qualitative Evaluation

We now validate how users prefer tags returned by TagAdvisor over writing reviews from scratch in a user study conducted on Amazon Mechanical Turk (AMT) on the real camera dataset. We also present an interesting anecdotal result returned by our algorithm for an entry in the real car dataset.

User Study: We conduct a user study through Amazon Mechanical Turk (AMT) to investigate if users prefer and benefit from our TagAdvisor system. We generate the top-$k$ tags for six cameras spanning different bands (Nikon, Canon, and Sony), and different types (digital SLR and compact point-and-shoot). The key objectives are: (i) to elicit the users’ responses to the tags returned by our system —if they find the tags meaningful and adequate to review the product or if they prefer articulating their own review; (ii) to elicit the users’ response to the products —if the feedback left by the users match the tags returned by our system.

We have 30 independent single-user tasks for each of the objectives. Each task is conducted in two phases: User Knowledge Phase and User Judgment Phase. During the first phase, we estimate the user’s familiarity about camera and digital photography in general, and the six cameras that are being reviewed. During the second phase, we collect responses to our questions in the study from the users who are estimated to have a reasonable background in the first phase. For the study involving the second objective, we consult domain experts to validate if the tags submitted by the users for the cameras are similar to the tags returned by our system.

Here are our observations.

- As many as 80% users confirmed that they have ever reviewed a product (or service) online, which is a high but understandable percentage since they are AMT workers – 75% of these users admitted that they do not write online reviews frequently.
- 67% of the users voted that they are knowledgeable about the six cameras (or, other similar cameras) that they have been asked to review in this study.

---

5www.mturk.com
• An overwhelming 83% of the users voted that they would submit online reviews more often if they are provided a set of meaningful keywords to choose from to express their feedback – 80% of these users clarified that their ‘Yes to TagAdvisor’ response is also dependent on what tags are provided to them for this purpose.

• 71% of the users reviewed the six cameras choosing tags returned by TagAdvisor instead of writing the review from scratch.

• Finally, 77% of the users submitted feedback that matches tags returned by TagAdvisor – 43% of those users submitted tags that are similar to the ones returned by the Independent Coverage problem while the rest 57% wrote tags that are similar to the ones returned by the Dependent Coverage problem, thereby endorsing that both Independent Coverage and Dependent Coverage problem are equally important.

• An interesting observation is that over 81% of users, who submitted their own tags wrote primarily about the more external aspects of the camera such as price, weight, physical look, lens, zoom, etc. instead of providing detailed comments about the quality of image, video capability, ease of use, etc. This is understandable since they are AMT workers and may not have used the exact same camera(s) in their recent past to provide in-depth feedback.

This validates the utility and usefulness of our system.

Case Study: We use the real car dataset to validate that our algorithms return meaningful tags - which meet user’s criticalness in reviewing, have sentiment attached to them, and also cover different aspects of the item - as opposed to the tags returned by existing tag recommendation systems [4, 10, 25]. Since [4] is the only tag recommender engine that returns tags that are relevant and diverse, we compare our result against it.

Suppose a user wants to submit her feedback for a 2010 Audi Q5 by choosing from a set of tags advised to her. If \( k = 6 \), the tags suggested by the tag recommender in [4] are:

amazing power, comfortable, convertible top with sunroof, nice style, good gas mileage, great auto transmission

Although this approach returns tags that cover diverse aspect of car, i.e, Standard Engine, Seats, Sunroof, Fuel Capacity, and Standard Transmission, it does not consider sentiment. All the 6 tags are positive.

Considering user factor parameter \( \alpha = 0.5 \), relevance parameter \( \beta = 0.5 \), our IC-TA algorithm returns the tags:

great auto transmission, good gas mileage, nice style, odd engine sound, wind noise at high speeds, uncomfortable rear seat

These tags not only covers same aspects of the car as above, i.e, Standard Transmission, Fuel Capacity, Standard Engine, Sunroof, and Seats, but it also satisfies the user’s criticalness in reviewing (\( \alpha = 0.5 \)), by returning three positive and three negative tags - the first three in the set above being positive and the last three being negative.

Under the same parameter specifications as above, our DC-TA algorithm returns the tags:

amazing power, convertible top with sunroof, comfortable, odd engine sound, wind noise at high speeds, uncomfortable rear seat

These tags not only cover different aspects of the car such as Standard Engine, Sunroof, and Seats but also allows the user to provide both positive and negative feedback for the same feature. Specifically, amazing power, odd engine sound are positive and negative tags respectively for the car feature Standard Engine. Two tags convertible top with sunroof, wind noise at high speeds are positive and negative tags for the car feature Sunroof. The last pair of tags comfortable, and uncomfortable rear seat are positive and negative tags for the car feature Seats. Thus, the user has the option to select positive and/or negative feedback about this feature when she submits her feedback.

6. RELATED WORK

Tag Recommendation: Tag recommendation has been extensively studied in literature [4, 10, 25, 13, 127]. The authors in [13] focused on user perspective and they proposed a probabilistic framework for solving the personalized tag recommendation, but without considering diversity. Result diversification has been studied in tag recommendation domain by [27, 4]; however, they take into account the possible topics and their goal is to provide high coverage and low redundancy with respect to those topics. The authors in [4] used the general probabilistic framework in [1] to address relevance and coverage. However, they assumes topics are independent, upon which a tag can not be dependent to the combination of the topics. The authors in [25] deals with the automated process to suggest useful and informative tags based on historical information. In our problem, the tags are more feedback than information about the resource and hence calls for additional properties like coverage of all item attributes as well as sentiment polarity in opinion of the user for the different attributes of the item. A recent work [10] proposes an optimization-based graph method for personalized tag recommendation. Though it considers both user features and item features for tag recommendation, the ranking-based solution recommends popular tags related to one or few specific aspects of the product and may evoke the rich-get-richer phenomenon, which in-turn is orthogonal to our objective of coverage. For example, if the popular tags for a point and shoot digital camera are lightweight, thin, and portable, the method would return them as the top tags even though they are all related to the weight of the product. We intend to return tags covering the different aspects of the product such as weight, price, etc. as well as the different sentiments in opinion such as light weight, heavy weight, low price, high price, etc. so that the user can submit her review objectively. The authors in [13] focused on user perspective and they proposed a probabilistic framework for solving the personalized tag recommendation, but without considering diversity.

Review Mining: There has been a considerable amount of work in review summarization, ranking and selection [11, 14, 16, 26]; yet, none of them can be readily extended to handle our problem. Review summarization creates statistical descriptions (i.e., a short snippet of text by extracting few existing sentences) of the review corpus in order to extract the proportion of positive and negative opinions about different aspects of a product. However, none of the current work directly caters to our objective of identifying personalized (i.e., user and item specific) tags. We leverage item descriptions, user demographics, as well as user sentiment. Review ranking aims to produce a score for each review...
and then display the top-k highest-scoring reviews to the user [11]. More specifically, [11] proposed two ranking mechanisms for ranking product reviews: consumer-oriented ranking mechanism ranks the reviews according to their expected helpfulness, and a manufacturer-oriented ranking mechanism ranks the reviews according to their expected effect on sales. However, they do not seek coverage over the range of features that are important to users and hence may return redundant information. For example, the top reviews for a point and shoot digital camera may just mention how ultrathin and portable it is, and not mention anything about how it has poor battery life. Review summarization identifies a subset of helpful reviews that collectively provide both the negative and the positive aspects of each commented feature [26]. While these methods do manage to expand the coverage of features and hence, diversify, they fail to capture the statistical properties of the actual review corpus. For example, if majority of the reviews for a SLR digital camera mention how excellent video quality it produces, that should be given higher weight than returning one positive and one negative opinion about the camera feature video quality. While [16] returns a characteristic set of reviews that respects the proportion of opinions on each feature (both positive and negative), as observed in the underlying corpus, neither does it leverage user preferences (demographics, sentiment, etc.), nor does it leverage user feedback for other similar items - both of which are necessary considerations of the set of tags returned by our problem.

Rule Learning: In this paper, we used existing techniques to find the rules of the complex dependencies among item attributes and the tags. Rule learning has been extensively studied and there are different techniques such as: rule based classifiers techniques like RIPPER [6, 17, 21], learning-based techniques like Re-RX [8] [23]. In rule base classifiers, rules can be extracted directly from data [17, 6] or it can be extracted from other classification models [21]. In [17], association rule mining is used to extract the rules while in [6] rules are extracted sequentially and for one class at a time. The authors in [20] describe a technique for transforming decision trees to succinct collection of if-then rules. Authors in [2] studied how to reduce the number of final rules in decision tree; [24] proposed a new method that can integrate rules from multiple trees in a random forest to improve the comprehensiveness of the extracted rules. There has been many prior work on extracting classification rules from Support Vector Machines (SVM) [19], [7], [3], and [8]. In [19] rules are extracted from ellipsoids and hyper-rectangles formed using clustering algorithms. The fuzzy rule extraction method [7] utilizes trained SVs to generate rule from each SV for each class.

7. CONCLUSION

In this paper, we introduce the novel TagAdvisor problem that leverages available user feedback for items in online review sites to simplify the review writing task. Our framework returns top-k tags relevant to the product a user is reviewing, have sentiment attached to them, and cover the diverse attributes of the product. To the best of our knowledge, our framework is the first to consider all three measures simultaneously in the context of tag mining. Our work is also the first to address the popular problem in the web - how to motivate users to review a product online - in a principled way. We formulate the problem as a general-constrained optimization goal. By adopting different definitions of coverage, we identify two concrete problem instances that enable a wide range of real-world scenarios. We show that these problems are NP-hard and develop practical algorithms with theoretical bounds to solve them efficiently. Our experiments validate the utility of our problem and demonstrate that our proposed solutions generate equally good quality results as exact brute-force algorithms with much less execution time. In the future, we plan to handle updates and insertions of new users, items and feedback. We also intend to evaluate the applicability of this framework to other applications, e.g., how to recommend hashtags to users in Twitter such that they can share their opinions, interests and comments for their topics of interest.

8. ACKNOWLEDGMENT

The work of Azade Nazi and Gautam Das was partially supported by National Science Foundation under grants 0915834, 1018865, Army Research Office under grant W911NF-15-1-0020 and a grant from Microsoft Research. Any findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors listed above.

9. REFERENCES

10. APPENDIX

10.1 Proof of Theorem 5

The proposed heuristic DC-TA algorithm 2 produce a solution with 2-approximation of the optimal, i.e. $\vartheta_{DC}(T'_{opt}) \leq 2 \vartheta_{DC}(T_{opt})$.

Proof. The proof is by induction on $k$. When $k \leq 2$ there is two possible cases, i) $k_1 = k_2 = 1$ ii) $k_1 = 0, k_2 = 2$ or $k_1 = 2, k_2 = 0$. Without loss of generality, we assume $\vartheta_{DC}(T_{opt})$ is positive. It is obvious that the statement is true after the first addition because in those cases an edge of minimum weight or maximum weight is selected respectively. Suppose that $k > 2$ and the statement holds for $k - 2$. We prove it holds for $k$ as well.

Assume $e = (v_{t_1}, v_{t_2})$ be an edge in graph $G'_{DC-TA}$ be the induced graph by $V'_2 \setminus v_{t_1}, v_{t_2}$, and $T'_{grk-2}$ and $T''_{optk-2}$ be the approximation and optimal solution for graph $G'_{DC-TA}$ with $k = 2$ nodes. Let us also assume that the end points of e’ be two tags in $T''_{optk-2}$. There are three different possibilities for the e’. i) $e' = e$ where $t_i, t_j \in T''_{optk-2}$ ii) one of the end points of the e be in $T''_{optk}$ so $e' = (v_{t_1}, v_{t_2})$ or $e' = (v_{t_2}, v_{t_1})$ where $t_2 \in T''_{optk} \setminus opt_{k-2}$; iii) $t_i, t_j \notin T''_{optk}$ and $e' = (v_{t_1}, v_{t_2})$ where $t_k, t_l \in T''_{optk} \setminus opt_{k-2}$. To have $T''_{grk} = T''_{grk-2} \cup \{t_i, t_j\}$ there are two possible case base on the selected number of tags from each partition. 

Case 1. From $k$ tags, $k_1 - 2$ and $k_2$ positive and negative tags are selected as $T'_{k_1-2}$ and $T'_{k_2-2}$ respectively. Thus, $t_i, t_j \in T^+$ and we assume that $e = (v_{t_i}, v_{t_j})$ be the edge with maximum weight in $T^+$. It is clear in all three different possibilities of e’ that $w(e) \geq w(e')$. In $T''_{grk}$, there are $2(k_1 - 2)$ number of edges from $T'_{k_1-2}$ and $2k_2$ edges from $T'_{k_2-2}$ which are incident to e. By triangle inequality those edges can contribute at least $k_1w(e) - (k_1 - 2)w(e)$ to $\vartheta_{DC}(T''_{grk})$ and those edges incident to e’ have total weight of at most $1/2k_2w(e) - 1/2(k_1 - 2)w(e)$. Using the induction hypothesis and the fact that $-w(e) \leq -w(e')$, we can show that gain of the $T''_{grk}$ is at most twice of the gain of the $T''_{optk}$.

$$\vartheta_{DC}(T''_{optk}) \geq -w(e') - 1/2(k_1 - 2)w(e) + 1/2k_2w(e) + \vartheta_{DC}(opt_{k-2})$$

$$\geq 1/2(-w(e) - (k_1 - 2)w(e) + k_2w(e)) + \vartheta_{DC}(T''_{grk})$$

$$\geq 1/2\vartheta_{DC}(T''_{grk})$$

Case 2. From $k$ tags, $k_1 - 1$ and $k_2 - 1$ positive and negative tags are selected as $T'_{k_1-1}$ and $T'_{k_2-1}$ respectively. Thus, $t_i \in T^+$ and $t_j \in T^-$ and we assume that $e = (v_{t_i}, v_{t_j})$ be the edge with minimum weight cross edge. It is clear that in all three different possibilities of e’, $w(e) \leq w(e')$. Since each end point of the e belong to each partition, the total contribution of e to $T''_{grk}$ would be the total number of cross cutting edges between $T'_{k_1-1}$ and $T'_{k_2-1}$ called $\omega(V_{T'_{k_1-1}}^{+}, V_{T'_{k_2-1}}^{-})$ minus the total weight of edges incident to $v_{t_i}$ and $v_{t_j}$ in each partition. Assume $t_i \in T'_{k_1-1}^{+}$ and by triangle inequality, $w(v_{t_i}, v_{t_j}) \leq w(e) + w(v_{t_j}, v_{t_i})$. Thus, $\sum_{u \in T'_{k_1-1}^{+}} w(v_{t_i}, v_u) = \sum_{u \in T'_{k_1-1}^{+}} w(v_{t_i}, v_u) - \sum_{u \in T'_{k_1-1}^{+}} w(v_{t_i}, v_u)$ would contribute at least $k_1w(e)$ to $T''_{grk}$ and at most $1/2k_1w(e)$ to $T''_{optk}$. Similar logic can apply to the nodes in $T'_{k_2-1}^{-}$. Using the induction hypothesis and the fact that $w(e) \leq w(e')$, we can show that gain of the $T''_{grk}$ is at most twice of the gain of the $T''_{optk}$.

$$\vartheta_{DC}(T''_{optk}) \geq w(e') + w(v_{t_i}, v_{t_j}) - w(v_{t_i}, v_{t_j})_{k_1-1} - w(v_{t_i}, v_{t_j})_{k_2-1} + \vartheta_{DC}(opt_{k-2})$$

$$\geq w(e') + \sum_{u \in T'_{k_1-1}^{+}} w(v_{t_i}, v_u) - \sum_{u \in T'_{k_1-1}^{+}} w(v_{t_i}, v_u) + \sum_{u \in T'_{k_1-1}^{+}} w(v_{t_i}, v_u)$$

$$\geq 1/2 \vartheta_{DC}(T''_{grk})$$

\[\square\]