Efficient Summarization Framework for Multi-Attribute Uncertain Data

Jie Xu, Dmitri V. Kalashnikov, and Sharad Mehrotra
Dept. of Computer Science, University of Irvine
Irvine, California, USA
jiex@uci.edu, dvk@ics.uci.edu, sharad@ics.uci.edu

ABSTRACT

This paper studies the problem of automatically selecting a small subset of representatives from a set of objects, where objects: (a) are multi-attributed with each attribute corresponding to different aspects of the object and (b) are associated with uncertainty – the problem that has received little attention in the past. Such object set leads to new challenges in modeling information contained in data, defining appropriate criteria for selecting objects, and in devising efficient algorithms for such a selection. We propose a framework that models objects as a set of the corresponding *information units* and reduces the summarization problem to that of optimizing probabilistic coverage. To solve the resulting NP-hard problem, we develop a highly efficient greedy algorithm, which gains its efficiency by leveraging *object-level* and *iteration-level* optimization. A comprehensive empirical evaluation over three real datasets demonstrates that the proposed framework significantly outperforms baseline techniques in terms of quality and also scales very well against the size of dataset.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval

Keywords

summarization; uncertain data; multi-attributes

1. INTRODUCTION

Summarizing data sets is a widely used technology to address the problem of user information overload [32, 20, 24]. For instance, Google Picasa supports a photo album overview functionality that provides a summary of pictures in a personal photo set. Review sites like Yelp often summarize data to provide highlights of opinions about different features of products/services, alleviating the user the burden of having to read numerous similar reviews. Summarization is likewise used in a variety of other application contexts (e.g., social media such as twitter, web search results, document sets, newspaper articles, etc.). Summarization is useful whenever users may wish to get a quick overview of a much larger set of objects.

Summaries, in general, can be categorized as either *abstractive* or *extractive*. While abstractive summaries synthesize information in a data set into a short precise synopsis, extractive summaries select a subset of objects (or part of objects) from the entire set as its representatives. Automated techniques to generate both forms of summaries have been extensively studied in the literature for a variety of domains [20, 22, 16, 31, 32, 10, 24]. Our focus in this paper is on extractive summaries.

Abstractly, it could be useful to view the process of extracting summaries as being composed of the following two steps: (1) Identify and associate with each object in a set of *"important (semantically meaningful) concepts"*, and (2) Select a subset of objects from the set as the summary that *"best represents"* the concepts contained in the set. For example, [20, 22] models the text summarization problem by mapping each sentence as a set of pre-defined or automatically learned topics, or N-grams, from which the best sentences that together cover most of the concepts in the documents are selected. Likewise, in the image domain, approaches such as [32, 16] model each image as a set of low-level visual features (edge histogram descriptor, color descriptor, etc.) or high-level visual concepts ("sky", "people", "indoor", etc.) and select representative images based on an optimization criteria. In the customer review domain, techniques like [10, 24] model each review as different opinions (i.e., positive, negative, neutral) on features of the product (e.g., ambiance, food quality, price for a restaurant set) and selects representative reviews based on such a conceptual model of data. While an extensive, and growing, body of work exists on summarization in diverse contexts and domains, existing approaches exhibit several limitations:

Conceptual Modeling of Objects. Most summarization techniques view individual objects as *single dimensional* entities – as a set of concepts generated usually by applying a single homogeneous type of analysis (e.g., extracting topics from textual content, N-gram based representation, image visual analysis) to associate semantic concepts with objects. However, in real-life applications, objects/entities, in general, are multidimensional and multi-attributed, amenable to diverse types of analysis. Consider the task of summarizing a set of Facebook or Picasa images for instance. Fig. 1 illustrates a sample picture from the domain. A picture contains visual concepts such as low and high level features extractable from visual content of the image. However, it may also be associated with information such as location/time of where/when

This work is part of the UCI Sherlock Project (http://sherlock.ics.uci.edu) on data quality and is funded in part by NSF grants IIS-1118114, CNS 1063596, CNS-1059436 and by DHS grant 206-000070.

SIGMOD'14, June 22–27, 2014, Snowbird, UT, USA.

http://dx.doi.org/10.1145/2588555.2588580.

Visual Concepts: sky – 0.7, water– 0.5, plant – 0.8, people – 0.9
Face Tags: " Kate", "Jeff"
Event: wedding
Time: 12/01/2012
Location: Los Angeles

Figure 1: A photo from Facebook.

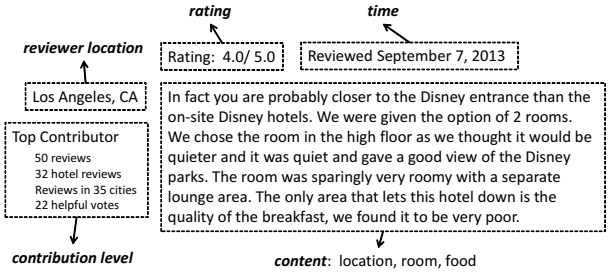


Figure 2: A review from TripAdvisor.

the image has been taken, which is often associated as meta-data with pictures. A picture might have more attributes, like names of individuals in the image. These name tags could be assigned by the user or could be automatically extracted using face recognition and/or other analysis techniques. Such real-world objects are better represented using multiple attributes. For instance, an image above could be modeled using five attributes: *visual concepts*, *face tags*, *events*, *time* and *location*.

Fig. 2 shows another example from a customer review set where the summarization process should consider objects (i.e., reviews) as consisting of multiple distinct attributes: *overall rating*, *timestamp*, *location* of the reviewer and *contribution level* of the reviewer, in addition to the attribute that captures and represents features extracted from textual content.

Multi-attribute representation of objects, besides being a more natural representation of real-world entities/objects, also exposes additional semantics information that should be considered by summarization approaches. For instance, some attributes like *face tags* are multi-valued and may exhibit strong associations between its values. For instance, "Kate" and "Jeff" may appear together in not just one image, but in multiple pictures. Likewise, correlations may also exists across multiple attributes, e.g., "Jeff" could be closely associated with the "Wedding" event. Such intra- and inter-attribute correlations are indeed semantic knowledge contained in the object set. Hence, they should be explicitly considered by a summarization technique for possible inclusion in a summary. Being able to identify and summarize such correlations is thus both a new challenge as well as an opportunity for summarization algorithms.

Selecting the "Best" Summary. Given a set of objects modeled as a set of concepts, summarization algorithms choose a good representative summary by optimizing some properties of summaries. Many such properties have been explored in the literature. They include *quality* of a representative selected [32] (modeled as efficiency of a chosen representative in [26]), *diversity* of objects in the summary [33], and so on. The predominant of all of such properties is *coverage* [20, 22]. Most work [4, 20, 22, 32] focus on maximizing coverage of the information in the set as the primary optimization criteria, while incorporating notions such as quality, efficiency, diversity as additional constraints.

Coverage, however, is defined over deterministic data wherein there is no uncertainty associated with the attribute values. If the attribute values are derived automatically using data analysis techniques (such as topic modeling [29], vision processing [12], or data cleaning [27, 28, 14]), they often contain uncertainty. For instance, visual concept detection technique like [12] applied to the photo in Fig. 1 may generate probabilistic concepts like "sky" with probability 0.7 (meaning the image contains sky with probability 0.7), "water" with probability 0.5, and so on.

Uncertainty in object attributes makes the summarization problem even more challenging. First, simply using the top-1 or threshold-based approach to convert probabilistic attributes into deterministic ones and optimizing deterministic *coverage* property will not work well and is largely dependent on the quality of techniques that generated the probabilities (we will show this in Section 6). This observation is consistent with recent work such as [19] which have shown such approaches to converting probabilistic data to deterministic representations to feed into higher level information processing / queries as sub-optimal. Second, optimization of *coverage* even without uncertainty in object attributes is NP-Hard [22, 32]. With uncertainty, to properly measure the *coverage* property, one needs to consider all possible worlds of attribute values of objects in the set, which makes the optimization problem intractable. We address this challenge by developing a *probabilistic coverage* model for which a very efficient greedy algorithm is then devised, which also provides an approximation bound.

Overall, the main contributions of this paper are:

- Summarization framework for multi-attribute data (which could contain uncertainty). The summarization framework, in addition to considering semantic content extracted by data analysis algorithms, also summarizes information such as correlations that are latent in the data set.
- Approach for *modeling* information contained in the multiple uncertain attributes of objects. Based on the model, we propose a probabilistic coverage property for an effective summarization of uncertain multi-attribute data. Summarization can thus be defined as an optimization of the coverage property (Sections 2, 3 and 4).
- Effective greedy algorithm, with an approximation bound, to the coverage optimization problem (which is NP-hard). The algorithm gains its high efficiency by leveraging *object-level* and *iteration-level* optimization techniques (Section 5).
- Extensive empirical study to demonstrates that the proposed summarization framework reaches high-quality results, and that it is very efficient, and scales very well against the size of data set (Section 6).

2. MODELING THE PROBLEM

In this section, we propose a method for modeling the extractive summarization problem in the context of data objects with multiple uncertain attributes. We first define the *object model* in a multi-attribute space in Section 2.1, followed by the approach for modeling of *information units* contained in the multiple uncertain attributes of objects in Section 2.2. Based on the model of information units, we propose the *probabilistic coverage* property for an effective summary in Section 2.3 and then model the summarization problem as an optimization of the coverage property in Section 2.4.

2.1 Modeling Data Objects

In the setup of the traditional extractive summarization problem, a summarization algorithm is given a set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$ to summarize. Though in theory objects might have multiple attributes, traditionally summarization is performed

only over one of them: let us call this attribute A. We refer to the domain of attribute A as $D(A) = \{v_1, v_2, \cdots, v_n\}$, which consists of a set of discrete values. We call each discrete value $v \in D(A)$ an *elemental value*. Let $o.A$ be the *attribute value* of object o in attribute A. As will be explained shortly, an *attribute value* can be different from an *elemental value*, namely $o.A$ could be a set of elemental values.

To be able to handle more complex objects used by modern applications, like the ones illustrated in Figs. 1 and 2, we will need to extend the traditional model of an object in a number of ways:

1. *Multi-attribute space.* Summarization will be done over a set of *multiple* attributes $\mathcal{A} = \{A_1, A_2, \cdots, A_{|\mathcal{A}|}\}$ instead of one. For instance, the object in Fig. 1 has five attributes: concepts, tags, album, time, and location.
2. *Single-valued and Multi-valued attributes.* To design a more generic solution, we will consider not only single-valued attributes, but also multi-valued attributes. For a *single-valued* attribute A, such as the *location* attribute in Fig. 1, the *attribute value* of an object is a single elemental value v from $D(A)$, thus $o.A = v$, for some $v \in D(A)$. In contrast, an attribute A is *multi-valued* if the *attribute value* of an object o in A is a set whose elements are elemental values from $D(A)$. Thus, $o.A = V$, for some set $V \subseteq D(A)$. For example, the attribute *tags* in Fig. 1 is multi-valued. If we refer to the photo as object o, then $o.tags = \{\text{Kate}, \text{Jeff}\}$ consisting of two elemental values from the domain of the *tags* attribute. For clarity and consistency of presentation, instead of writing $o.A = v$ for a single-valued attribute A, we will write $o.A = \{v\}$. Here, $\{v\}$ is a set but with a single elemental value v.
3. *A special-case.* The framework will be able to account for a special-case where the attribute value of an object o is not specified or unknown. For any single-valued or multi-valued attribute A, this will be denoted as $o.A = \emptyset$.
4. *Uncertain attributes.* In addition to deterministic attributes, our framework will also be able to handle uncertain attributes. Given an uncertain attribute A and an object o, each elemental value $v \in D(A)$ has a probability $P(v, o.A)$, or just $P(v, o)$ when it is clear from the context, that $v \in o.A$. For example, for a single-valued attribute *location*, a text extraction techniques could be uncertain and produce a location as a combination of, say, two mutually exclusive values NY with probability 0.8 and LA with probability 0.2. Similarly, for a multi-valued attribute, a visual concept detection technique like [12] may produce a set of visual concepts, each associated with a probability. For the example in Fig. 1, the values and associated probabilities in the attribute *concepts* are sky:0.7, water:0.5, plant:0.8 and people:0.9. The ground truth/gold standard for the image, of course, might contain only some subset of $\{\text{sky}, \text{water}, \text{plant}, \text{people}\}$. Note that elemental values in a multi-valued attribute are not necessarily independent.

We will use *feature vector* $F_o = (o.A_1, o.A_2, \cdots, o.A_{|\mathcal{A}|})$ to represent the attribute values associated with object o. Notation $F_o(A)$ will refer to the attribute value of object o in attribute A, that is, $F_o(A) = o.A$. Let us consider object o in Fig. 1 as an example. If we assume that the set of visual concepts associated with object o is $o.concept = \{\text{sky}, \text{plant}, \text{people}\}$, then the feature vector of o is $F_o = (\{\text{sky}, \text{plant}, \text{people}\}, \{\text{Kate}, \text{Jeff}\}, \{\text{wedding}\}, \{12/01/2012\}, \{\text{Los Angeles}\})$.

Since some attributes could be uncertain, the actual feature vector of an object can also be uncertain. We will use the notation $\mathcal{W}_o$, or just $\mathcal{W}$ when it is clear from the context, to denote the space/world of all possible feature vectors for object o. For instance, Fig. 3 lists all 12 possible feature vectors $\mathcal{W}_o = \{F_1, F_2, \ldots, F_{12}\}$ for the multi-attribute space. The single-valued attribute *location* could take only one of three attribute values: $\emptyset$, {LA},or {NY}. Naturally, the location cannot be LA and NY at the same time as these are mutually exclusive values for the domain. The multi-valued attribute *face* can take one of the following four attribute values: $\emptyset$, {Kate}, {Jeff}, and {Kate, Jeff}.

	o.location	*o*.face
F_1	∅	∅
F_2	∅	{Kate}
F_3	∅	{Jeff}
F_4	∅	{Kate, Jeff}
F_5	{LA}	∅
F_6	{LA}	{Kate}

	o.location	*o*.face
F_7	{LA}	{Jeff}
F_8	{LA}	{Kate, Jeff}
F_9	{NY}	∅
F_{10}	{NY}	{Kate}
F_{11}	{NY}	{Jeff}
F_{12}	{NY}	{Kate, Jeff}

$U(F_8)$
{LA}
{Kate}
{Jeff}
{Kate, Jeff}
{LA, Kate}
{LA, Jeff}
{LA, Jeff, Kate}

Figure 3: All possible feature vectors for object o for the case when the multi-attribute space consists of only two attributes: *location* and *face*. Here, $D(location) = \{\text{LA}, \text{NY}\}$ and $D(face) = \{\text{Kate}, \text{Jeff}\}$. $U(F_8)$ is the set of IUs for feature vector F_8.

The way to generate the multi-attribute space and the attribute values of objects depends on the data domain and the application. Consider a photo set from social media such as Facebook. The proposed multi-attribute space for the set has four attributes: *visual*, *event*, *face* and *time*. Using visual features proposed by [12], an SVM classifier can be trained for generating visual concepts in the *visual* attribute, such as "sky", "water", "people", etc. a personal event ontology benchmark proposed by [23] can be used to define categories (wedding, graduation, picnic, etc.) in the *event* attribute. The *face* attribute denotes the people tagged in the photos. The *time* attribute is obtained by clustering the time stamps of the photos in the set. Among the four attributes, only the *time* attribute is *single-valued*.

2.2 Modeling Information Unit

Extractive summarization techniques choose a small subset S of objects from the entire object set $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$ to represent $\mathcal{O}$. One of the key ideas in this process is that the information contained in some object $o \in \mathcal{O}$ could be *covered* by summary S even if object o is not included in S. A simple example is where, say, an object set contains several photos of *Jeff* in *Los Angeles*. Then including just one such photo in the summary can cover the remaining ones – as the user will be able to get the fact that *Jeff* was in *Los Angeles* from that one photo.

In general, the goal of our summarization framework is to generate a summary that maximally covers the information in the whole set. Before we formalize the *coverage* property of a summary, we first model the information contained in objects in the context of the multi-attribute space. We model each object o as a container (i.e., a set) of *information units* associated with the object. Intuitively, an *information unit* (IU) is a basic fact defined in the multi-attribute space. It could be a fact representing an individual elemental value or a fact capturing correlation among multiple elemental values. For instance, an image that contains "Jeff" in "Los Angeles" will have "Jeff" and "Los Angeles" as its IUs. In addition, if correlation between "Jeff" and "Los Angeles" is interesting, this correlation will be another IU.

To capture elemental values as well as correlation, we will represent each single IU as a set of elemental values $U = \{v_1, v_2, \cdots, v_{|U|}\}$,

In this paper we use correlation in its general sense of "a mutual relationship or connection between two or more things". It is different from the traditional concept of correlation coefficient.

where each $v_i \in U$ is an elemental value from $D(A)$ for some $A \in \mathcal{A}$. We will consider IUs of three categories and associate each object with three sets of IUs:

1. *Set of Elemental IUs.* For a feature vector $F_o = (o.A_1, o.A_2, \cdots, o.A_{|\mathcal{A}|})$ of object o, the set of elemental IUs, $U_{elem}(F_o)$, represents all the elemental values of F_o:

$$U_{elem}(F_o) = \{\{v\} : \exists A \in \mathcal{A} \text{ s.t. } v \in o.A\}.$$

For example, for F_8 in Fig. 3, $U_{elem}(F_8) = \{\{\text{LA}\}, \{\text{Jeff}\}, \{\text{Kate}\}\}$. In turn, the set of elemental IUs for a given entire attribute A is defined as $U_{elem}(A) = \{\{v\} : v \in D(A)\}$.

2. *Set of Intra-Attribute Correlations.* Recall that an attribute value in a *multi-valued* attribute is a set of elemental values. Some of these elemental values could be correlated. For example, if "Jeff" and "Kate" often appear in photos together, then the *face* attribute can have an interesting (intra-attribute) correlation that consists of these two values. In other words, the group of these two people shown together in the photo is an interesting concept and should be also captured as an IU. Similarly, if "Jeff" and "Kate" are present together in just one or very few images, such a correlation might not be interesting. Hence, we can define the set of *interesting* intra-attribute correlations $U_{intra}(A)$ for a given multi-valued attribute A. We will explain how to compute this set in Section 3. Notice, for any *single-valued* attribute A it holds that $U_{intra}(A) \equiv \emptyset$.
For a feature vector $F_o = (o.A_1, o.A_2, \cdots, o.A_{|\mathcal{A}|})$ of object o, $U_{intra}(F_o)$ represents the set of all interesting *intra-attribute correlation* IUs contained specifically in F_o:

$$U_{intra}(F_o) = \{U : \exists A \in \mathcal{A} \text{ s.t. } U \in U_{intra}(A) \wedge U \subseteq o.A\}.$$

For the example in Fig. 3, assuming all intra-attribute correlations are interesting, we have $U_{intra}(F_8) = \{\{\text{Jeff}, \text{Kate}\}\}$.

3. *Set of Inter-Attribute Correlations.* Like intra-attribute correlations, correlations of elemental values across different attributes can also be of interest when summarizing data. For example, for feature-vector F_8 from Fig. 3, the IUs of object o for the *location* attribute include {LA} and {NY} (elemental IUs). For the *face* attribute, the IUs include {Jeff}, {Kate} (elemental IUs), and {Jeff, Kate} (intra-attribute correlation IU). In addition to the above IUs, the user may also be interested in capturing joint concept across a set of different attributes, such as "*Jeff* is in *LA*" and "*Jeff* and *Kate* are together in *LA*". To represent such joint concepts, we use the third type of IU, referred to as *inter-attribute correlation*.
To simplify the formal definition, let us assume that each elemental value also carries which attribute it belongs to, so that $D(A_i) \cap D(A_j) = \emptyset$ for all $i \neq j$. In other words, we do not want to mix "Jeff" that is an elemental value in the *face* attribute with "Jeff" that is an elemental value in another attribute, say, *tags*, etc. Instead, we treat them, as two different values: "face.Jeff" and "tags.Jeff". We refer to the set of interesting inter-attribute correlations given the multi-attribute space $\mathcal{A}$ as $U_{inter}(\mathcal{A})$. We will explain how to compute $U_{inter}(\mathcal{A})$ in Section 3. For a feature vector $F_o = (o.A_1, o.A_2, \cdots, o.A_{|\mathcal{A}|})$ of object o, $U_{inter}(F_o)$ represents the set of all interesting *inter-attribute correlations* contained specifically in F_o:

$$U_{inter}(F_o) = \{U : U \subseteq \bigcup_{A \in \mathcal{A}} o.A \wedge U \in U_{inter}(\mathcal{A})\}.$$

This definition simply states that an IU of this type, U, is a subset of elemental values of F_o that belongs to the class of interesting inter-attribute correlations. For the example in Fig. 3, assuming all inter-attribute correlations are interesting, we have $U_{inter}(F_8) = \{\{\text{LA}, \text{Jeff}, \text{Kate}\}, \{\text{LA}, \text{Kate}\}, \{\text{LA}, \text{Jeff}\}\}$.

objects	F_O (Location)	F_O (Face)
O_1	{LA}	{Kate, Jeff}
O_2	∅	{Kate}
O_3	{NY}	{Jeff}

Figure 4: One possible world $\mathcal{G}$ of feature vectors taken by objects $\{o_1, o_2, o_3\}$.

Let $U(F_o)$ denote the set of all IUs for F_o, that is, $U(F_o) = U_{elem}(F_o) \cup U_{intra}(F_o) \cup U_{inter}(F_o)$. For the case from Fig. 3, assuming all intra-attribute and inter-attribute correlations are interesting, all 6 information units that form $U(F_8)$ are listed in Fig. 3.

To make the summarization framework more generic, we allow for each information unit U to be associated with a non-negative weight w_U, which reflects the importance of U. Be default, we assume each IU is equally important and thus all $w_U = 1$. However, depending on the application domain, the user might want to set these weights differently. For instance, the user might set it to $\frac{|U|}{|\mathcal{A}|}$, which will represent the fraction of non-null values in U and thus reflects how specific is a given U. Our framework is generic and imposes no restriction on how to set these weights.

2.3 Probabilistic Coverage Model

The goal of summarization is to maximize the recipient's knowledge about the information in the entire set. Thus, having defined the notion of *information unit*, we are now able to define the *coverage* property of a summary. Assume that we are given a data set $\mathcal{O} = \{o_1, o_2, \cdots, o_{|\mathcal{O}|}\}$ and a summary $\mathcal{S} \subseteq \mathcal{O}$ extracted from the set. Similar to coverage properties used in various contexts [4, 20, 22, 32], the *coverage* property of the summary, denoted as $Cov(\mathcal{S}, \mathcal{O})$, reflects the proportion of the information contained in the whole set $\mathcal{O}$ covered by summary $\mathcal{S}$. Since attribute values associated with objects are uncertain, in this section we propose a probabilistic extension to the coverage model.

Given a multi-attribute space $\mathcal{A}$ and a data set $\mathcal{O}$, we denote as $\mathcal{W}^{|\mathcal{O}|}$ the space of all possible worlds of feature vectors of objects in $\mathcal{O}$. Then by $\mathcal{G}(o) = F_o$ we will denote the feature vector that corresponds to an object $o \in \mathcal{O}$ in world $\mathcal{G} \in \mathcal{W}^{|\mathcal{O}|}$. Consider Fig. 3 as an example, since there are 12 different feature vectors for one object, for a dataset with $|\mathcal{O}|$ objects, there will be totally $12^{|\mathcal{O}|}$ different worlds in $\mathcal{W}^{|\mathcal{O}|}$.

Assume that $\mathcal{G} \in \mathcal{W}^{|\mathcal{O}|}$ is the actual (ground truth/gold standard) world for the data set $\mathcal{O}$. *Notice, the algorithm, of course, does* not *know $\mathcal{G}$, and we will discuss how to deal with it shortly.* For any subset $S \subseteq \mathcal{O}$, we denote as $U_{\mathcal{G}}(S) = \bigcup_{o \in S} U(\mathcal{G}(o))$ the set of information units contained in S. Thus, given world $\mathcal{G}$, the coverage of summary $\mathcal{S}$ with respect to $\mathcal{O}$ is defined as the proportion of information in $\mathcal{O}$ covered by summary $\mathcal{S}$:

$$Cov_{\mathcal{G}}(\mathcal{S}, \mathcal{O}) = \frac{\sum_{o \in \mathcal{O}} \sum_{U \in U_{\mathcal{G}}(\{o\}) \cap U_{\mathcal{G}}(\mathcal{S})} w_U}{\sum_{o \in \mathcal{O}} \sum_{U \in U_{\mathcal{G}}(\{o\})} w_U} \quad (1)$$

Intuitively, the numerator of Eq. (1) reflects the amount of information in set $\mathcal{O}$ that is covered by summary $\mathcal{S}$, while the denominator reflects the total amount of information in set $\mathcal{O}$.

Example 1. Fig. 4 illustrates a possible world $\mathcal{G}$ of feature vectors for the set of 3 objects $\mathcal{O} = \{o_1, o_2, o_3\}$ from the multi-attribute space used in Fig. 1. For clarity of presentation, all information units in this example have the same weight of 1. As-

suming all *intra-attribute* and *inter-attribute* correlations are interesting, $U_{\mathcal{G}}(\{o_1\})$ includes 7 IUs as shown in Fig. 3. Thus, $|U_{\mathcal{G}}(\{o_1\})| = 7$. It is easy to see that $|U_{\mathcal{G}}(\{o_2\})| = 1$ and $|U_{\mathcal{G}}(\{o_3\})| = 3$. Assuming the selected summary is $\mathcal{S} = \{o_1\}$, then $|U_{\mathcal{G}}(\{o_1\}) \cap U_{\mathcal{G}}(\mathcal{S})| = 7$, $|U_{\mathcal{G}}(\{o_2\}) \cap U_{\mathcal{G}}(\mathcal{S})| = 1$ and $|U_{\mathcal{G}}(\{o_3\}) \cap U_{\mathcal{G}}(\mathcal{S})| = 1$. Therefore, $Cov_{\mathcal{G}}(\mathcal{S}, \mathcal{O}) = (7 + 1 + 1)/(7 + 1 + 3) = 9/11$.

Since the algorithm does not know $\mathcal{G}$, it cannot compute the coverage using Equation (1) directly. Instead, it uses *probabilistic coverage* which it can compute directly. Specifically, the (probabilistic) *coverage* $Cov(\mathcal{S}, \mathcal{O})$ is modeled as the expected *coverage* of summary $\mathcal{S}$ with respect to $\mathcal{O}$:

$$Cov(\mathcal{S}, \mathcal{O}) = \mathbb{E}(Cov_{\mathcal{G}}(\mathcal{S}, \mathcal{O})) = \sum_{\mathcal{G} \in \mathcal{W}^{|\mathcal{O}|}} \mathbb{P}(\mathcal{G}) \cdot Cov_{\mathcal{G}}(\mathcal{S}, \mathcal{O}) \tag{2}$$

where $\mathbb{P}(\mathcal{G})$ is the probability of world $\mathcal{G}$. It is, however, infeasible to directly use the expected value of $Cov_{\mathcal{G}}(\mathcal{S}, \mathcal{O})$ as the coverage of summary $\mathcal{S}$. That will require enumerating all possible worlds in $\mathcal{W}^{|\mathcal{O}|}$, which is exponential. We will discuss how to solve the efficiency issue of the probabilistic coverage model in Section 4.

2.4 Summarization Problem

With the above described probabilistic extension to coverage, we can model the summarization problem as follows:

Definition 1. (Summarization). Let $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$ be a set of objects (e.g. photo set, or set of consumer reviews). Given a positive number $K \in N^+$, where $K \leq |\mathcal{O}|$, the goal of the summarization problem is to find a set $\mathcal{S} \subseteq \mathcal{O}$ of size $|\mathcal{S}| = K$, such that $Cov(\mathcal{S}, \mathcal{O})$ is maximized.

In other words, the goal of the *summarization* task is to find the optimal summary $\mathcal{S}^\star$ determined as:

$$\mathcal{S}^\star = \underset{\mathcal{S} \subseteq \mathcal{O}, |\mathcal{S}| = K}{\operatorname{argmax}} \; Cov(\mathcal{S}, \mathcal{O}) \tag{3}$$

Having modeled the summarization problem, we are left with the following challenges unanswered:

1. How to choose *intra-* and *inter-attribute* correlations that are *interesting* (Section 3)?
2. How to efficiently compute the probabilistic coverage (Section 4)?
3. How to efficiently solve the optimization problem (Section 5)?

In the subsequent sections we present solutions that address these challenges.

3. IDENTIFYING INTERESTING CORRELATIONS

As mentioned in Section 2.2, interesting information pieces could exist not only in the form of elemental attribute value, but also as intra- and inter-attribute correlations. However, the challenge remains of deciding which intra- and inter-attribute correlations are interesting for summarization. We resolve this problem by mapping it into the *market basket problem*, where an equivalent challenge of finding interesting item-sets has been extensively studied in the past [5, 8, 17, 34].

To see the relation between our problem and that of interesting item-set mining, let us first briefly review the problem of interesting item-set mining. Let $\mathcal{D}$ be a set of all items. An item-set I is a subset of $\mathcal{D}$. Let $\mathcal{T}$ be a set of transactions where each transaction $T \in \mathcal{T}$ is an item-set. Given a set of items $\mathcal{D}$ and a set of transactions $\mathcal{T}$, the goal of interesting item-set mining is to find a set of *interesting* item-sets $\mathcal{I}$, where the *interestingness* of an item-set is measured using certain metrics, such as frequency [5] and correlation [34].

Consider as an example a correlation measure $corr(I)$ which measures the correlation between items in I. Many different correlation measures have been proposed in the past, such as *All-confidence*, *Max-confidence*, *Cosine*, *Coherence*, *Kulczynski*, and so on [34]. For example, Kim et al. [17] use *Cosine* measure defined as:

$$Cosine(I) = \frac{sup(I)}{\sqrt[|I|]{\prod_{i \in I} sup(\{i\})}}. \tag{4}$$

where $sup(I)$ is the support of an item-set I, defined as the number of transactions in $\mathcal{T}$ that contain all items in I.

Observe that the goal of the problem of finding interesting intra-attribute correlations $U_{intra}(A)$, that we need to solve, is very similar for the deterministic case. Namely, we are given a deterministic multi-valued attribute A and a set of objects $\mathcal{O} = \{o_1, o_2, \cdots, o_{|\mathcal{O}|}\}$. Each object $o \in \mathcal{O}$ is associated with a set of elemental values $o.A \subseteq D(A)$. Each correlation is represented as a set of elemental values $V \subseteq D(A)$. By mapping $D(A)$ into the set of items $\mathcal{D}$ and $o.A$ for one object $o \in \mathcal{O}$ into one transaction $T_{o.A} \in \mathcal{T}$, we can map the problem of finding interesting intra-attribute correlations into that of mining interesting item-sets and thus solve it using existing techniques. Our framework does not impose any limitation on the technique of interesting item-set mining, and different techniques could be used, e.g., [5, 8, 17]. By default, we employ the correlation measure defined in Eq. (4) and the correlated item-set finding algorithm proposed in [8].

However, the above techniques are deterministic and cannot be applied "as is" to non-deterministic (uncertain) attributes we are considering. Recall that for an uncertain attribute, each elemental value $v \in D(A)$ is associated with a probability $P(v, o)$ indicating the likelihood that value v is associated with object o. To solve the above issue, we employ the expected item-set support proposed in [7] instead of the regular $sup(I)$ function, while applying the technique of mining correlated item-set over precise dataset.

Interesting inter-attribute correlations $U_{inter}(\mathcal{A})$, can also be selected in a way similar to that for selecting intra-attribute correlations. Given a multi-attribute space $\mathcal{A} = \{A_1, A_2, \cdots, A_{|\mathcal{A}|}\}$ and a set of objects $\mathcal{O}$, we view each elemental IU or intra-attribute correlation IU as an item, that is, we map $\bigcup_{A \in \mathcal{A}}(U_{elem}(A) \cup U_{intra}(A))$ into $\mathcal{D}$ and $U_{elem}(F_o) \cup U_{intra}(F_o)$ for one object $o \in \mathcal{O}$ into one transaction $T \in \mathcal{T}$. Therefore, like intra-attribute correlation, we can select interesting inter-attribute correlations using techniques of interesting item-set mining, with one constraint that each mined item-set must consists of items from unique attributes. In the resulting set of interesting item-sets $\mathcal{I}$, each item-set $I \in \mathcal{I}$ corresponds to either an elemental IU or an intra-attribute correlation. Thus, each inter-attribute correlation $U \in U_{inter}(\mathcal{A})$ can be constructed as $U = \bigcup I$ for each $I \in \mathcal{I}$.

4. EFFICIENTLY COMPUTING COVERAGE MODEL

As mentioned in Section 2.3, the probabilistic coverage defined in Eq. (2) can not be computed efficiently simply by enumerating an exponential number of possible worlds. In this section we propose an alternate way of computing probabilistic coverage that achieves the same goal, but can be computed efficiently.

Assume that $\mathcal{G}$ is the ground truth world for $\mathcal{O}$, that is, the true deterministic combination of elemental values. Notice, $\mathcal{G}$ is, of

course, unknown in general due to uncertainty. Then we can define the total amount of information in data set $\mathcal{O}$ covered by summary $\mathcal{S}$, denoted $Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O})$, as:

$$Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O}) = \sum_{o \in \mathcal{O}} \sum_{U \in U_{\mathcal{G}}(\{o\}) \cap U_{\mathcal{G}}(\mathcal{S})} w_U. \quad (5)$$

Given a dataset $\mathcal{O}$ and any positive constant $C_{\mathcal{O}}$ that is independent of $\mathcal{S}$, from Eq. (1) it can be easily derived that:

$$\operatorname*{argmax}_{\mathcal{S} \subseteq \mathcal{O}, |\mathcal{S}|=K} Cov_{\mathcal{G}}(\mathcal{S}, \mathcal{O}) = \operatorname*{argmax}_{\mathcal{S} \subseteq \mathcal{O}, |\mathcal{S}|=K} Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O})/C_{\mathcal{O}}.$$

Therefore, our goal reduces to that of maximizing the value of $Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O})/C_{\mathcal{O}}$.

Our framework will use the notion of probability $P(U, o)$ that object o contains an information unit U, however the framework is independent from how $P(U, o)$ is computed. For example, under the assumption of independence between attributes, $P(U, o)$ can be computed as $P(U, o) = \prod_{v \in U} P(v, o)$, where $P(v, o)$ is the probability that object o takes elemental value v. If the independence assumption does not hold, more advanced techniques for computing $P(U, o)$, e.g. [15], could be used in our framework as well.

Therefore, assuming independence between objects, the probability $P(U, \mathcal{S})$ that information unit U is covered by summary $\mathcal{S}$ can be computed as the probability that it is covered by at least one object in $\mathcal{S}$:

$$P(U, \mathcal{S}) = 1 - \prod_{o \in \mathcal{S}} (1 - P(U, o)). \quad (6)$$

Using this formula, we now can prove the following lemma:

Lemma 1. Given a summary $\mathcal{S}$, the expected amount of information in data set $\mathcal{O}$ covered by $\mathcal{S}$ can be computed as:

$$\mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O})) = \sum_{o \in \mathcal{S}} \sum_{U \in \mathcal{U}} P(U, o) \cdot w_U + \\ + \sum_{o \in \mathcal{O} \setminus \mathcal{S}} \sum_{U \in \mathcal{U}} P(U, o) \cdot P(U, S) \cdot w_U \quad (7)$$

where $\mathcal{U} = \{U | \exists o \in \mathcal{O}, P(U, o) > 0\}$ is the set of all possible information units contained in objects in $\mathcal{O}$.

Proof. Intuitively, the first term of Eq. (7) corresponds to the coverage of information units in the summary. The second term corresponds to the information units in objects other than the summary that are covered by the summary. The detailed proof is in [3]. □

Thus, for probabilistic coverage $Cov(\mathcal{S}, \mathcal{O})$ it holds that:

$$Cov(\mathcal{S}, \mathcal{O}) = \mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O}))/C_{\mathcal{O}}, \quad (8)$$

where $C_{\mathcal{O}} = \sum_{o \in \mathcal{O}} \sum_{U \in \mathcal{U}} P(U, o) \cdot w_U$. Notice that $Cov(\mathcal{S}, \mathcal{O})$ takes values in $[0, 1]$. The *summarization* problem thus reduces to generating the summary $\mathcal{S}$ that maximizes $Cov(\mathcal{S}, \mathcal{O})$ defined in Eq. (8).

5. GREEDY APPROACH TO SUMMARIZATION

It can be proven by a reduction from the Maximum Coverage problem that maximizing $Cov(\mathcal{S}, \mathcal{O})$ is NP-Hard [3]. Since no efficient exact solutions to NP-hard problems are known, in this section we develop a basic greedy solution to the summarization problem, which has an approximation bound. To make our approach even more efficient, we propose two levels of efficiency optimizations for the greedy algorithm.

Notation	Meaning
$\mathcal{O}$	the whole data set
$\mathcal{S}$	summary of $\mathcal{O}$
K	size of the final summary generated by the algorithm
$\mathcal{A}$	multi-attribute space
$\mathcal{U}$	all possible information units given $\mathcal{D}$
$\mathcal{G}$	ground truth world for elemental values of objects in $\mathcal{O}$
$Cov(\mathcal{S}, \mathcal{O})$	coverage of summary S, with respect to dataset $\mathcal{O}$
$P(U, o)$	probability of object o to contain information unit U
$P(U, \mathcal{S})$	probability of unit U to be covered by summary $\mathcal{S}$
w_U	weight of information unit U

Table 1: Notation.

5.1 Submodularity of Probabilistic Coverage

To develop the greedy algorithm with approximation bound, we first show the submodularity property of the proposed probabilistic *coverage* model.

Definition 2. (Submodularity). Given a finite set $\mathcal{O}$, a set function $f : 2^{|\mathcal{O}|} \to \mathcal{R}$ is submodular iff for any two sets $\mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{O}$ and $o \in \mathcal{O} \setminus \mathcal{T}$ it holds that $f(\mathcal{S} \cup \{o\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{o\}) - f(\mathcal{T})$.

That is, for a submodular f, adding o to a smaller subset of a set will have larger incremental impact on f than adding o to the set itself. The connection between submodularity and coverage of a summary is very intuitive: adding an object to a small summary will have larger incremental impact compared to that of the same summary with many more objects added. We now formally prove the following lemma.

Lemma 2. $Cov(\cdot, \mathcal{O})$ is a non-decreasing submodular set function.

Proof. According to Eq. (7), for any $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{O}$, $\mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}', \mathcal{O})) - \mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O}))$ can be computed as

$$\sum_{o \in \mathcal{S}' \setminus \mathcal{S}} \sum_{U \in \mathcal{U}} P(U, o) \cdot (1 - P(U, \mathcal{S})) \cdot w_U \\ + \sum_{o \in \mathcal{O} \setminus \mathcal{S}'} \sum_{U \in \mathcal{U}} P(U, o) \cdot (P(U, \mathcal{S}') - P(U, \mathcal{S})) \cdot w_U$$

It can be seen from Eq. (6) that $1 \geq P(U, \mathcal{S}') \geq P(U, \mathcal{S})$. As a result, $\mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}', \mathcal{O})) - \mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O})) \geq 0$. Since $Cov(\mathcal{S}', \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) = \mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}', \mathcal{O})) - \mathbb{E}(Total_{\mathcal{G}}(\mathcal{S}, \mathcal{O})) / C_{\mathcal{O}} \geq 0$, $Cov(\cdot, \mathcal{O})$ is a non-decreasing set function.

Let us now prove the submodularity of $Cov(\cdot, \mathcal{O})$. It is trivial to derive that:

$$Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) = [\sum_{U \in \mathcal{U}} P(U, o)(1 - P(U, \mathcal{S}))w_U \\ + \sum_{o' \in \mathcal{O} \setminus (\mathcal{S} \cup \{o\})} \sum_{U \in \mathcal{U}} P(U, o')(P(U, \mathcal{S} \cup \{o\}) - P(U, \mathcal{S}))w_U]/C_{\mathcal{O}} \quad (9)$$

It is easy to see that the term $1 - P(U, \mathcal{S})$ is non-increasing as $|\mathcal{S}|$ increases. It can also be derived that $P(U, \mathcal{S} \cup \{o\}) - P(U, \mathcal{S}) = P(U, o)(1 - P(U, \mathcal{S}))$. Thus, $P(U, \mathcal{S} \cup \{o\}) - P(U, \mathcal{S})$ is non-increasing. As a result, $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O})$ is non-increasing. Therefore, for any two sets $\mathcal{S}, \mathcal{T} \subseteq \mathcal{O}$ such that $\mathcal{S} \subseteq \mathcal{T}$ and $o \notin \mathcal{T}$, $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) \geq Cov(\mathcal{T} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{T}, \mathcal{O})$. We can now conclude that $Cov(\cdot, \mathcal{O})$ is a non-decreasing submodular set function. □

In the context of non-decreasing submodular set functions f where $f(\emptyset) = 0$, it has been proven [25] that the greedy algorithm that always selects the object whose addition maximizes the function f

GREEDY-BASIC$(K, \mathcal{O})$
1 $\mathcal{S} \leftarrow \emptyset$
2 **while** $|\mathcal{S}| < K$ **do**
3 $o^\star \leftarrow \text{argmax}_{o \in \mathcal{O} \setminus \mathcal{S}}$ *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$
4 $\mathcal{S} \leftarrow \mathcal{S} \cup \{o^\star\}$
5 **return** $\mathcal{S}$

Figure 5: Basic Greedy Algorithm for Summarization.

produces a solution with approximation ratio $(1 - 1/e)$, where e is the base of the natural logarithm. That is, the greedy algorithm yields a solution that is at least $(1 - 1/e)$ fraction of the optimal solution. Therefore, the following lemma holds:

Lemma 3. The greedy algorithm for the problem of maximizing *Cov*$(\mathcal{S}, \mathcal{O})$ has approximation ratio $(1 - 1/e)$. [25]

5.2 Basic Greedy Algorithm

Let us now describe the greedy algorithm in detail and explore how to implement it efficiently. The outline of the algorithm is shown in Fig. 5. It starts with an empty summary $\mathcal{S}$. At every step of the greedy process, the algorithm traverses the remaining objects in $\mathcal{O} \setminus \mathcal{S}$ and adds into $\mathcal{S}$ the object $o^\star$ whose addition will lead to maximum *coverage* of $\mathcal{O}$ by the resulting set $\mathcal{S} \cup \{o^\star\}$.

The largest contribution to the computational complexity of the greedy algorithm comes from the **while** loop (Steps 2-4). In each iteration, it needs to compute *coverage* *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$ resulted from adding each object $o \in \mathcal{O} \setminus \mathcal{S}$ into $\mathcal{S}$ (Step 3). Let us analyze the complexity of each iteration at two levels:

1. **Object-level***: time spent computing coverage for one object.* Based on Eq. (6), it takes $O(K)$ to compute $P(U, \mathcal{S} \cup \{o\})$ for each information unit $U \in \mathcal{U}$. As a result, based on Eqs. (7) and (8), it takes $O(|\mathcal{O}| \cdot K \cdot |\mathcal{U}|)$ to compute the *coverage* *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$ for one object $o \in \mathcal{O} \setminus \mathcal{S}$.
2. **Iteration-level***: number of object-level operations in each iteration.* The algorithm needs to compute the *coverage* for $|\mathcal{O} \setminus \mathcal{S}|$ objects in each iteration.

Since there are totally K iterations, the overall complexity of the basic greedy algorithm is $O(K^2 \cdot |\mathcal{O}|^2 \cdot |\mathcal{U}|)$. Although this naive implementation of the greedy algorithm takes polynomial time, the complexity is still high for large datasets. We next explore how to further optimize the algorithm at the two levels: *object-level* and *iteration-level*.

5.3 Object-level Optimization

From Fig. 5, we can see that in each iteration of the greedy process, the basic greedy algorithm tries to add a new object $o^\star$ to the summary $\mathcal{S}$ that is generated so far, such that the new summary $\mathcal{S} \cup \{o^\star\}$ has maximum *coverage* *Cov*$(\mathcal{S} \cup \{o^\star\}, \mathcal{O})$. The goal of object-level optimization is to reduce the complexity of computing *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$ for one object o. The basic idea of *object-level* optimization is that instead of directly computing *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$ in each iteration, we can compute the value based on *Cov*$(\mathcal{S}, \mathcal{O})$ from the last iteration. The main question is how to do this computation very efficiently.

To efficiently compute *coverage* *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$, one approach is to employ Eq. (9) to update *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$ based on the value of *Cov*$(\mathcal{S}, \mathcal{O})$. However, it can be seen from Eq. (9) that the complexity of this approach is $O(|\mathcal{O}| \cdot |\mathcal{U}|)$, which is actually inefficient in practice. We propose an efficient algorithm to update the value of *coverage*. Let us first define a concept $C(U, o) = P(U, o) \cdot w_U$ that indicates the contribution of information unit U to the *coverage* score of object o. The contribution of information unit U to the

GREEDY-OBJECT-LEVEL-OPT$(K, \mathcal{O})$
1 $\mathcal{S} \leftarrow \emptyset$ // initialize the summary
2 $C_\mathcal{O} \leftarrow \sum_{o \in \mathcal{O}} \sum_{U \in \mathcal{U}_o} P(U, o) \cdot w_U$ // pre-computed once
3 **for each** $U \in \bigcup_{o \in \mathcal{O}} U_o$ **do** // for each U contained in $\mathcal{O}$
4 $c_U \leftarrow C(U, \mathcal{O})$ // initialize contribution of U
5 $p_U \leftarrow 0$ // initialize probability of U being covered by $\mathcal{S}$
6 **for each** $o \in \mathcal{O}$ **do**
7 *covDif*$_o \leftarrow$ *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$ − *Cov*$(\mathcal{S}, \mathcal{O})$ // computed as Lemma 4
8 *cov*$_o \leftarrow$ *covDif*$_o$ // initialize variable *cov*$_o$: coverage of $\mathcal{S} \cup \{o\}$
9 **return** GREEDY-PROCESS$(\mathcal{S}, K, \mathcal{O})$

Figure 6: Greedy Algorithm for Summarization with *object-level* optimization.

GREEDY-PROCESS$(\mathcal{S}, K, \mathcal{O})$
1 *create object* $o^\star$ *such that* *cov*$_{o^\star} = 0$
2 **while** $|\mathcal{S}| < K$ **do**
3 **for each** $o \in (\mathcal{O} \setminus \mathcal{S})$ **do** // **for** *loop is* $\mathcal{O}(|\mathcal{O} \setminus \mathcal{S}| \cdot |\mathcal{U}|)$
4 *cov*$_o \leftarrow$ *cov*$^\star_o + \sum_{U \in \mathcal{U}_o} P(U, o) \cdot (1 - p_U)$
$\cdot (w_U + c_U - C(U, o))/C_\mathcal{O}$ // update *cov*$_o$ using Eq. (11)
5 $o^\star \leftarrow \text{argmax}_{o \in \mathcal{O} \setminus \mathcal{S}}$ *cov*$_o$
6 $\mathcal{S} \leftarrow \mathcal{S} \cup \{o^\star\}$
7 **for each** $U \in U_{o^\star}$ **do**
8 $p_U \leftarrow 1 - (1 - p_U) \cdot (1 - P(U, o^\star))$ // update p_U
9 $c_U \leftarrow c_U - C(U, o^\star)$ // update c_U
10 **return** $\mathcal{S}$

Figure 7: Greedy Process with *object-level* optimization.

coverage score of set of objects $\mathcal{T}$ is then defined as:

$$C(U, \mathcal{T}) = \sum_{o \in \mathcal{T}} C(U, o) \tag{10}$$

Then the following lemma holds (see proof in [3]):

Lemma 4. For any summary $\mathcal{S} \subseteq \mathcal{O}$ and any object $o \in \mathcal{O} \setminus \mathcal{S}$, *Cov*$(\mathcal{S} \cup \{o\}, \mathcal{O})$ − *Cov*$(\mathcal{S}, \mathcal{O})$ can be computed as:

$$\frac{\sum_{U \in U_o} P(U, o) \cdot (1 - P(U, \mathcal{S})) \cdot (w_U + C(U, \mathcal{O} \setminus (\mathcal{S} \cup \{o\})))}{C_\mathcal{O}}$$

where $U_o = \{U | U \in \mathcal{U}, P(U, o) > 0\}$ is the set of information units that are associated with object o with non-zero probabilities.

Based on Lemma 4, we can update coverage as follows:

$$\textit{Cov}((\mathcal{S} \cup \{o^\star\}) \cup \{o\}, \mathcal{O}) = [\sum_{U \in U_o} P(U, o) \cdot (1 - P(U, \mathcal{S} \cup \{o^\star\})) \cdot (w_U + C(U, \mathcal{O} \setminus (\mathcal{S} \cup \{o^\star\} \cup \{o\})))]/C_\mathcal{O} + \textit{Cov}(\mathcal{S} \cup \{o^\star\}, \mathcal{O}) \tag{11}$$

As shown in Fig. 7, the algorithm with *object-level* optimization maintains the following variables at the end of every iteration of the greedy process:

1. $p_U = P(U, \mathcal{S} \cup \{o^\star\})$ for each information unit $U \in \mathcal{U}$, which is the probability of information unit U covered by the summary after adding the best object $o^\star$. It can be derived from Eq. (6) that $P(U, \mathcal{S} \cup \{o^\star\}) = 1 - (1 - P(U, \mathcal{S})) \cdot (1 - P(U, o^\star))$. Thus, at the end of each iteration, for each information unit $U \in U_{o^\star}$, p_U can be updated in constant time (Step 8).
2. $c_U = C(U, \mathcal{O} \setminus (\mathcal{S} \cup \{o^\star\}))$ for each information unit $U \in \mathcal{U}$, which is the contribution of information unit U to the coverage score of the objects in $\mathcal{O} \setminus (\mathcal{S} \cup \{o^\star\})$. According to Eq. (10), $C(U, \mathcal{O} \setminus (\mathcal{S} \cup \{o^\star\})) = C(U, \mathcal{O} \setminus \mathcal{S}) - C(U, o^\star)$. Therefore, for one information unit $U \in U_{o^\star}$, c_U can be updated in constant time in each iteration (Step 9).

3. $cov_o = Cov(\mathcal{S} \cup \{o\}, \mathcal{O})$ for each object $o \in \mathcal{O}$. Since p_U and c_U can be updated in constant time, according to Eq. (11), cov_o can be updated in $O(|U_o|)$.

The whole greedy algorithm with *object-level* optimization is shown in Fig. 6, which includes the preprocess (Steps 1-8) that initializes the above variables (the variable $covDif_o$ can be ignored for now and will be explained shortly) and the greedy process (Step 9).

Lemma 5. The computation complexity of the greedy algorithm with *object-level* optimization is $O(K \cdot |\mathcal{O}| \cdot |U_o|)$, where $|U_o|$ is the average number of information units associated with objects with non-zero probabilities. (See proof in [3].)

Therefore, compared to the complexity of the basic greedy algorithm, *object-level* optimization saves a factor of $K \cdot |\mathcal{O}| \cdot |\mathcal{U}|/|U_o|$.

5.4 Iteration-level Optimization: pruning

The efficiency of the greedy algorithm can be further improved by *iteration-level* optimization. Recall that in each iteration, the greedy algorithm tries to add an object $o \in \mathcal{O} \setminus \mathcal{S}$ to summary $\mathcal{S}$, such that the *coverage* $Cov(\mathcal{S} \cup \{o\}, \mathcal{O})$ is maximized. As shown above, it requires $O(|U_o|)$ to update $Cov(\mathcal{S} \cup \{o\}, \mathcal{O})$ for one object o. The goal of *iteration-level* optimization is to reduce the number of operations for updating coverage scores in each iteration.

Score difference. Before describing the *iteration-level* optimization, let us first introduce the concept *score difference* used in the algorithm. Given a summary $\mathcal{S}$ and an object o to be added into $\mathcal{S}$, *score difference* $Dif(\mathcal{S}, o)$ is defined as as follows:

$$Dif(\mathcal{S}, o) = Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) \quad (12)$$

Therefore, instead of *coverage* $Cov(\mathcal{S} \cup \{o\}, \mathcal{O})$ for each object $o \in \mathcal{O} \setminus \mathcal{S}$, we can compute and maximize $Dif(\mathcal{S}, o)$. In other words,

$$\operatorname*{argmax}_{o \in \mathcal{O} \setminus \mathcal{S}} Dif(\mathcal{S}, o) = \operatorname*{argmax}_{o \in \mathcal{O} \setminus \mathcal{S}} Cov(\mathcal{S} \cup \{o\}) \quad (13)$$

Using *score difference* also provides an efficient way to prune operations for computing $Dif(\mathcal{S}, o)$ as will be seen as follows.

Pruning. The idea is to efficiently compute an *upper bound* value $Upper(\mathcal{S}, o)$ of $Dif(\mathcal{S}, o)$ for each object $o \in \mathcal{O} \setminus \mathcal{S}$. As shown in the Fig. 8, if the upper bound value $Upper(\mathcal{S}, o)$ is less than the best value of $Dif(\mathcal{S}, o^\star)$ found so far, the algorithm can prune the computation of $Dif(\mathcal{S}, o)$ for this object (Steps 6).

Upper Bound. The question is how to compute the *upper bound* for each object $o \in \mathcal{O} \setminus \mathcal{S}$. The goal is to compute an upper bound in constant time. Let us first state the following lemma.

Lemma 6. Given any two summaries $\mathcal{S}$ and $\mathcal{S}'$, such that $\mathcal{S}' \subset \mathcal{S} \subseteq \mathcal{O}$. For any object $o \in \mathcal{O} \setminus \mathcal{S}$, $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) \leq \min(Cov(\mathcal{S}' \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}', \mathcal{O}), 1 - Cov(\mathcal{S}, \mathcal{O}))$.

Proof. According to Lemma 2, $Cov(\mathcal{T}, \mathcal{O})$ is a submodular set function over $\mathcal{T}$. Thus, $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) \leq Cov(\mathcal{S}' \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}', \mathcal{O})$. By the definition of coverage in Eq. (8), $Cov(\mathcal{T} \cup \{o\}, \mathcal{O}) \leq 1$ for any summary $\mathcal{T}$. Also $Cov(\mathcal{T}, \mathcal{O})$ is a non-decreasing set function over $\mathcal{T}$ according to Lemma 2. Therefore, $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) \leq 1 - Cov(\mathcal{S}, \mathcal{O})$. Hence, we can conclude that $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O}) \leq min(Cov(\mathcal{S}' \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}', \mathcal{O}), 1 - Cov(\mathcal{S}, \mathcal{O}))$. □

Variable	Meaning
cov_o	$Cov(\mathcal{S} \cup \{o\}, \mathcal{O})$, for summary $\mathcal{S}$ in current iteration
p_U	$P(U, \mathcal{S})$, for summary $\mathcal{S}$ in current iteration
c_U	$C(\mathcal{U}, \mathcal{O} \setminus \mathcal{S})$, for summary $\mathcal{S}$ in current iteration
$covDif_o$	$Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O})$, difference of coverage scores between two iterations

Table 2: Variables in each iteration of greedy algorithms

GREEDY-PROCESS-PRUNE($\mathcal{S}, K, \mathcal{O}, o^\star$)
1 *create object* $o^\star$ *such that* $cov^\star_o = 0$
2 **while** $|\mathcal{S}| < K$ **do**
3 $o' \leftarrow o^\star$, $dif^\star \leftarrow 0$
4 **for each** $o \in (\mathcal{O} \setminus \mathcal{S})$ **do**
5 $upper \leftarrow \min(covDif_o, 1 - cov_{o'})$ // Eq. (14)
6 **if** $upper < f^\star$ **continue** // *prune*
7 $cov_o \leftarrow cov_{o'} + \sum_{U \in \mathcal{U}_o} P(U, o) \cdot (1 - p_U) \cdot (w_U + c_U - C(U, o))/C_{\mathcal{O}}$ // update cov_o using Eq. (11)
8 $covDif_o \leftarrow cov_o - cov'_o$ // update variable $covDif_o$
9 $dif \leftarrow covDif_o$
10 **if** $dif^\star < dif$ **do** // update best object $o^\star$
11 $dif^\star \leftarrow dif$, $o^\star \leftarrow o$
12 $\mathcal{S} \leftarrow \mathcal{S} \cup \{o^\star\}$
13 **for each** $U \in U_{o^\star}$ **do**
14 $p_U \leftarrow 1 - (1 - p_U) \cdot (1 - P(U, o^\star))$ // update p_U
15 $c_U \leftarrow c_U - C(U, o^\star)$ // update p_U
16 **return** $\mathcal{S}$

Figure 8: Greedy Process with *iteration-level* optimization.

Hence, given a summary $\mathcal{S} \subseteq \mathcal{O}$, for any $\mathcal{S}' \subset \mathcal{S}$, the upper bound value $Upper(\mathcal{S}, o)$ for $Dif(\mathcal{S}, o)$ can be computed as

$$Upper(\mathcal{S}, o) = \min(1 - Cov(\mathcal{S}, \mathcal{O}), Cov(\mathcal{S}' \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}', \mathcal{O})) \quad (14)$$

The algorithm maintains for each $o \in \mathcal{O}$ a variable $covDif_o$ that stores the value of $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O})$, where $\mathcal{S}$ is the summary generated in the iteration where $covDif_o$ is last updated. The variable $covDif_o$ is first computed in the pre-processing (Step 7, Fig. 6). In each iteration of the greedy process shown in Fig.8, if one object o is not pruned, $covDif_o$ will be updated (Step 8) towards a tight bound. Even if object o is pruned, $covDif_o$ is still a valid upper bound on $Cov(\mathcal{S} \cup \{o\}, \mathcal{O}) - Cov(\mathcal{S}, \mathcal{O})$ according to Lemma 6. Therefore, the upper bound value computed in Step 5 of Fig. 8 is still a valid upper bound for $Dif(\mathcal{S}, o)$. Based on the above analysis, for one object o, the value of upper bound for $Dif(\mathcal{S}, o)$ can be computed in constant time (Step 5).

Table 2 summarizes important variables used in *object-level* and *iteration-level* optimizations.

Lemma 7. The computational complexity of the greedy algorithm with both *object-level* and *iteration-level* optimization is $O(|\mathcal{O}| \cdot |U_o| + K \cdot |A| \cdot |U_o|)$, where $|A|$ is the average number of objects in each iteration that are not pruned. (See proof in [3].)

6. EMPIRICAL EVALUATION

6.1 Datasets

1. `Flickr Photo Set` consists of 20000 photos collected from photo sharing website Flickr and MIRFLICKR image dataset [11]. The multi-attribute space for this set includes 3 attributes: *visual*, *event* and *time*. The visual concepts (e.g., "sky", "water", "people", etc.) are generated by using an SVM classifier over visual features proposed in [12]. Personal event ontology benchmarks proposed by [23] are used

Dataset	# objects	# elemental IUs	# intra-attr	# inter-attr
Flickr	20000	153	65	748
Review	2495	43	19	120
Facebook	220	72	81	437

Table 3: Statistics about datasets

to define categories (wedding, graduation, picnic, etc.) in the *event* attribute. User generated textual tags (e.g., album names, photo descriptions etc) are used to associate photos to event categories. The *time* attribute is obtained by clustering the time stamps of the photos in the set. Among the three attributes, only the *time* attribute is *single-valued*.

2. `Review Dataset` consists of customer reviews about 10 hotels, collected from TripAdvisor and OpinRank Dataset [9]. Each hotel has about 250 reviews on average, which is a typical size of the datasets used in review summarization work [10, 24]. Reviews about the same hotel are treated as a set to be summarized. We represent information in each review in three attributes: *Rating*, *Facet*, and *Time*, where *rating* denotes the overall rating given to the hotel by the reviewer, and *facet* represents different aspects of the hotels mentioned in the review. We trained naive Bayes classifiers for 6 hotel facets: room, service, location, parking, facility and food. We also used Alchemy API [1] to do sentiment analysis that further classifies each hotel facet into three sub-facets: positive, negative and neutral. Thus, the *facet* attribute includes 18 elemental values in total. Elemental values of the *time* attribute are obtained by clustering the timestamps of reviews within the same set. Among the three attributes, only the *facet* attribute is *multi-valued*.
3. `Facebook Photo Set` consists of 220 photos uploaded by 10 Facebook users who have a common Facebook friend. We model this dataset as a multi-attribute space consisting of 4 attributes: *visual*, *event*, *time* and *face*. The first three attributes *visual*, *event*, and *time* are constructed in the same way as in the Flickr photo set. The *face* attribute denotes the people tagged in the photos by Facebook users. Since this set is not too large, we will use it to conduct a human evaluation on various approaches.

Table 3 lists important statistics about all the 3 datasets, including number of objects, number of distinct elemental IUs, number of distinct intra- and inter-attribute correlations in each dataset. For the review dataset, the number of elemental IU, intra- and inter-attribute correlations are averaged over the 10 hotels.

6.2 Performance Metrics

A straightforward way of evaluating summaries is to compare them with a human generated ground truth as is often done in literature related to summarizing text [21]. Such a human-evaluation approach is, however, very expensive for large data sets. Also, in some domains such as photo, summarization is a subjective task and is likely to vary across people. Thus, in addition to a human-based evaluation over a small set, to measure the performance of summarization approaches quantitatively, we employ two widely used automatic evaluation metrics:

Query-based coverage metric. To evaluate whether a summary covers the information in the data set, we defined a query based information model, similar to [32]. The goal of an effective summary is to cover important information units in the data set. Each object in the data set can be modeled as the set of information units in it. Given an information unit U and a candidate summary $\mathcal{S}$, we define a binary function $\mathit{InfoPres}(U, \mathcal{S})$, whose value is 1 if information U is present in summary $\mathcal{S}$ and 0 otherwise. Given a data set $\mathcal{O}$, based on the weight of an information unit U and its frequency in $\mathcal{O}$, the importance of U is modeled as $\mathit{Gain}(U, \mathcal{O}) = \frac{w_U \cdot |\{o: o \in \mathcal{O} \wedge U \in U_o\}|}{\sum_{o \in \mathcal{O}} \sum_{U' \in U_o} w_{U'}}$. Let $\mathcal{U}$ be the set of all IUs contained in the entire set $\mathcal{O}$, based on the ground truth elemental values of objects in $\mathcal{O}$. By viewing each $U \in \mathcal{U}$ as a query and $\mathit{Gain}(U, \mathcal{O})$ as the gain of satisfying this query, the performance of a candidate summary $\mathcal{S}$ of data set $\mathcal{O}$, can be evaluated by the overall gain of satisfying $U \in \mathcal{U}$, defined as follows:

$$\mathit{InfoGain}(\mathcal{S}, \mathcal{O}) = \sum_{U \in \mathcal{U}} \mathit{InfoPres}(U, \mathcal{S}) \cdot \mathit{Gain}(U, \mathcal{O}) \qquad (15)$$

Similar ideas of query-based models have been used in multiple domains, such as image [32] and text [4].

Representativeness. To evaluate whether the summary represents the distribution of information in the data set, we compare the distribution of information units in the summary with that in the data set. We use a smoothed version of Jensen Shannon (JS) divergence [2] to measure the distance between the two distributions of information units in the data set $\mathcal{O}$ and a candidate summary $\mathcal{S}$. Given a set of objects $\mathcal{T}$, let us first define the probability distribution $\mathbb{P}_\mathcal{T} : \mathcal{U} \rightarrow (0, 1]$ which assigns each information unit U in $\mathcal{T}$ a probability value $\mathbb{P}_\mathcal{T}(U)$:

$$\mathbb{P}_\mathcal{T}(U) = \frac{\epsilon + w_U \cdot |\{o : o \in T, U \in U_o\}|}{\epsilon \cdot |\mathcal{U}| + \sum_{o \in \mathcal{T}} \sum_{U' \in U_o} w_{U'}}, \ \forall U \in \mathcal{U}$$

where is ϵ is set to a small value of 0.0005 to avoid shifting the probability mass by a large amount to unseen events. KL divergence [2] between two distribution $\mathbb{P}$ and $\mathbb{Q}$ is defined as

$$D_{KL}(\mathbb{P}||\mathbb{Q}) = \sum_{U \in \mathcal{U}} \mathbb{P}(U) \cdot \ln[\mathbb{P}(U)/\mathbb{Q}(U)]$$

Thus, the representativeness of a summary $\mathcal{S}$ with respect to the data set $\mathcal{O}$ is defined as:

$$D_{JS}(\mathcal{S}, \mathcal{O}) = [D_{KL}(\mathbb{P}_\mathcal{S}||\mathbb{P}_\mathcal{O}) + D_{KL}(\mathbb{P}_\mathcal{O}||\mathbb{P}_\mathcal{S})]/2 \qquad (16)$$

6.3 Approaches

1) *Greedy.* This is the greedy algorithm presented in Section 5 with the probabilistic coverage model we proposed in Section 2.3.

2) *No correlation.* This approach is the same as the above approach, except that in optimizing the probabilistic coverage property, we do not consider any correlations (intra- or inter-attribute) in generating summaries. We will compare such an approach with the above greedy approach to highlight the importance of considering correlations in summarization.

3) *Clustering.* This approach uses K-means clustering to generate K clusters for the data set and selects from each cluster the object that is closest to the cluster centroid, where V_o is used as the feature vector for clustering. The clustering algorithm is used as another baseline.

4) *Random.* The random approache selects K objects (without replacement) randomly from the data set. Like the clustering approach, random also serves as a baseline.

6.4 Experiments

Experiment 1 (Performance of Greedy Algorithm). This experiment tests the performance of the proposed greedy algorithm by comparing it with two baselines: 'clustering' and 'random', in terms of both information gain and representativeness. Fig. 9 shows *InfoGain* scores for different approaches on the Flickr dataset. The

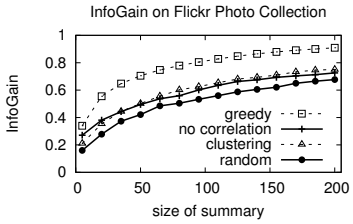


Figure 9: InfoGain on Flickr Photo Set.

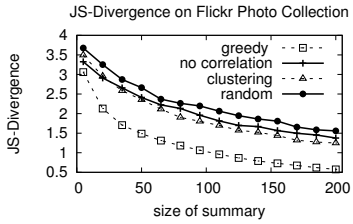


Figure 10: JS-Divergence on Flickr.

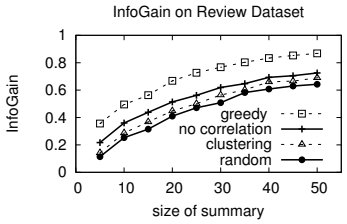


Figure 11: InfoGain on Reviews.

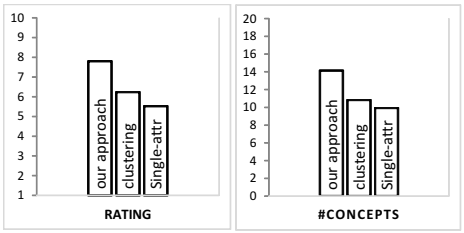


Figure 12: Human evaluation scores on Facebook photo set.

x-axis represents summary sizes and the y-axis represents the *InfoGain* score computed as Eq. (15). We can see from the figure that 'greedy' is much better than both 'clustering' and 'random'. Fig. 10 shows JS-Divergence of different approaches on Flickr Photo Set. Likewise, we can see a clear advantage of our approach ('greedy') over baselines, in terms of representativeness. One interesting observation from comparing Figs. 9 and 10 is that there is a strong correlation between *InfoGain* score and *JS-Divergence* score. It has been shown by [12] that the JS-Divergence based metric has very good correlation with human evaluation. Thus, both metrics are actually good indicators of human preference.

Similarly, we can see from Fig. 11 that our approach ('greedy') also outperforms baselines in terms of *InfoGain* on the review dataset.

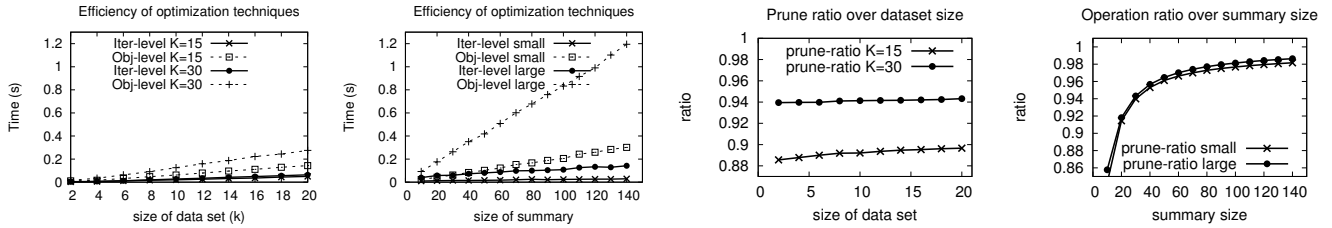
Experiment 2 (Importance of correlations). To see the importance of correlations (intra- and inter attribute), we compare our approach 'greedy' with 'no correlation' that ignores any correlations while optimizing the probabilistic coverage property. From Figs. 9 and 10, we can see that 'no correlation' has almost the same performance as 'clustering' and is worse than 'greedy'. The gap between 'greedy' and 'no correlation' illustrates that ignoring intra- and inter-correlations will lead to sub-optimal results. Similar results can be seen on the review dataset (Fig. 11). The only difference is that on the review dataset, 'no correlation' is slightly better than 'clustering'. Also, the gap between 'no correlation' and 'greedy' is smaller than that on Flickr photo set. This is because in the Flickr photo set, the ratio between the number of correlations and the number of elemental IUs is about 5:1, whereas in the review dataset the ratio is about 3:1 on average.

Experiment 3 (Human evaluation on Facebook photo set). In addition to the quantitative experiments above, we also conduct a human evaluation involving 50 people who are not related to the paper. To each person, we first show the Facebook photo set and then the summaries generated by various approaches. Each summary consists of 10 photos. We asked each evaluator to (a) rate each summary on the scale of 1 to 10, and (b) identify which of the 20 concepts randomly chosen from the set of concepts in the photos could be learnt from the summary. To identify the 20 concepts to probe, we first randomly sample one photo from the photo set, and then randomly sample one information unit contained in the object as the concept. Fig. 12 shows the rating and the number of checked concepts for all approaches involved in this experiment, averaged over all 50 evaluators. Our approach is better than 'clustering' in terms of both people's subjective rating and number of checked concepts. To further establish the advantage of considering multi-attribute space in summarization, we add an approach, referred to as *single-attr*, that tries to cover only information about photo content (*visual concept* attribute). The comparison between our approach and 'single-attr' illustrates that people prefer a summary that covers information in multiple attributes.

Experiment 4 (Efficiency of optimization techniques). This experiment tests the efficiency of the two optimization techniques we proposed: *object-level* and *iteration-level* optimizations, on an Intel i7 2GHz machine with 8G main memory. We test the efficiency on the Flickr photo set since it has large number of data objects. Fig. 13a shows the efficiency of the two optimization techniques with two summary sizes (K = 15 and K = 30) over different size of the data sets. The time spent by basic greedy algorithm is not shown in the figure, because it is much slower than the two optimizations techniques. It can be seen from the figure that the two optimizations techniques both run less than 0.3 second and scale well against the size of dataset. Compared to *object-level* optimization, applying *iteration-level* optimization technique makes the greedy algorithm even faster (within 0.1 second) and scales much better against size of dataset. The gap between the two optimizations techniques increases as dataset size increases. This is because *iteration-level* prunes away a lot of computations in each iteration of the greedy process. Another interesting observation is that while the time spent by *object-level* increases as the summary size is doubled (from K = 15 to K = 30), the change of *iteration-level* is invisible.

The above observation becomes clearer in Fig. 13b, which shows the efficiency of the two optimization techniques over different summary sizes, while the size of dataset is fixed at 6k and 20k. From this figure, we can see that the *iteration-level* optimization can still spend less than 0.2 second for large dataset size and summary size. It scales very well against summary size. That explains why *object-level* spends almost the same time for the two summary size: K = 15 and K = 30, as shown in Fig. 13a. By comparing Figs. 13a and 13b, we can see that the gap between the two optimization techniques increases more quickly with summary size than with dataset size.

Since the advantage of *iteration-level* optimization over *object-level* optimization is due to the pruning of computations in *iteration-level* optimization, we investigate more about the effect of the pruning. Recall that without pruning, every operation to update *coverage* (Fig. 7, Step 4) will be executed. We referred to as op_{total} the total number of such operations in the whole greedy process. With *iteration-level* optimization, a lot of such operations will be pruned (Fig. 8, Steps 6). We referred to as op_{prune} the number of pruned operations in the whole greedy process. We define *prune-ratio* as the proportion of pruned operations out of all operations: op_{prune}/op_{total}. Fig. 13c and 13d show the *prune-ratio* against dataset size and summary size respectively. From the two figures, we ob-



(a) Efficiency over size of dataset. (b) Efficiency over size of summary. (c) Ops-ratio over size of dataset. (d) Ops-ratio over size of summary.

Figure 13: Efficiency of optimization techniques

serve the prune-ratio achieved by the *iteration-level* optimization is from 0.86 to 0.98, which indicates that the upper bound (Eq. (14)) used in the algorithm is a tight bound. A comparison between the two figures reveals that the *prune-ratio* increases faster with summary size than with dataset size. The reason is that $Cov(\mathcal{S}, O)$ is a submodular set function over $\mathcal{S}$, thus as the summary size $|\mathcal{S}|$ increases, the upper bound defined in Eq. (14) gets tighter and tighter.

Other experiments. In addition to the experiments reported above, we have also done the following experiments. However, due to space limitations, we will only summarize the results of them.

1) *Comparison to non-probabilistic approach.* We compare our approach with existing summarization techniques that take as input only data with deterministic attributes. We convert the probabilistic dataset into a deterministic one. To do that, we use the most natural approach where values with probabilities > 0.5 are included to the object and the other ones are discarded. We then apply the photo summarization approach proposed in [32]. The result on the Flickr dataset shows that our approach is about 25% better than the non-probabilistic approach for small summary sizes (e.g. 20 - 50).

2) *Robustness.* Since the proposed summarization framework is designed to handle probabilistic attributes, a natural question arises of how our approach will be affected by the quality of data processing techniques that generate the probabilistic attributes. We measure the decline of the *InfoGain* of our approach as a function of noise levels added to the probabilities. The result shows that our approach is quite robust as the quality of probabilities gets worse.

7. RELATED WORK

In general, summarization can be categorized as extractive and abstractive. In this section, we focus on related work on extractive summarization, where the task is to choose a small fixed-size subset of objects to represent the entire set. The extractive summarization problem has been extensively studied in the past for a variety of data domains, settings, and summarization properties. Below we explore these issues in more detail.

Data domain. The problem of summarizing data sets has been investigated for various domains of data, including images, user reviews, text documents, microblogs. The generic framework we propose in this paper applies to these domains as well.

In the image domain, Kennedy et al. [16] propose a method to generate representative images of landmarks. Simon et al. [31] address the problem of scene summarization from online image sets. These works focus on image analysis such as visual feature based clustering and image similarity calculation, but not how to summarize images given the results of image analysis. However, our goal is to model the *coverage* property of a summary and efficiently generate the summary with optimized coverage property. Authors in [32] propose a summarization framework for personal photo sets. However, the coverage model in [32] can not be applied in such settings where photos have uncertain attribute values, while we propose a probabilistic coverage model to handle such uncertainty.

Another data domain where summarization plays an important role is the customer review domain. Hu et al. [10] aim to mine and summarize customer reviews of a product by mining frequently commented product features and sentiment analysis of opinion sentences. However, they focus on product feature identification and sentiment analysis rather than summary generation. The product review summarization system in [24] summarizes sentences about the same product feature by clustering the sentences and selecting the most representative sentence in each cluster. In the context of objects with multiple uncertain attributes, applying such a cluster-based approach generates sub-optimal results in terms of information coverage, as demonstrated in Section 6.4.

The summarization problem has also been studied in text document / micro-blogging domain. Li et al. [20] define *coverage* as the amount of subtopics of the document covered by the summary. The document summarization method proposed in [22] aims to maximize the coverage of the summary defined as the overall benefits gained from the terms contained in the summary. These techniques consider only the deterministic single-value case. Inouye et al. [13] propose a tweet summarization approach that first generates K clusters using k-means and select from each cluster the best tweet. Again, in terms of information coverage, the cluster-based approach generates sub-optimal result.

Yang et al. [35] propose a method for summarizing relational databases by clustering and ranking relational tables by their importance. Their goal is to summarize the relational tables instead of the data tuples in the tables, while our goal is to summarize data objects. Also, they focus on defining table importance and similarity between tables, while we focus on the summarization framework. Their technique cannot be directly applied to solve our problem.

Application setting. Summarization has been applied in various application settings. We categorize these application settings as static and streaming. In a static setting, the goal is to provide a summary of a given data set. Examples include personal photo set [32], community-contributed web photo set [16], customer reviews of individual product [24], single text document [20], etc. All these works do not consider multiple uncertain attributes associated with data objects to be summarized. The authors in [33] study the problem of computing diverse query results in the context of online shopping applications, where diversity is measured based on a lexicographical ordering of attributes. However, their technique cannot be applied in our problem setting, because in their work (a) the notion of coverage is suppressed; (b) a strict lexicographical preference must be known in advance, which is not a requirement for the generic multi-attribute space considered in our paper.

In a streaming setting, the data set to be summarized is changing as time evolves. Shou et al. [30] explores the problem of tweet stream summarization. It focuses on tweet stream clustering algorithm and data structure to enable historical summarization for any time interval. These works are complementary to ours. While we focus on summarizing static data, our summarization approach

can be applied in streaming setting for summarizing data in a given time interval.

Summary property. A variety of summary properties have been identified in different contexts/domains. Authors in [32] identify the *quality* as an important property, which indicates the appeal of the photos in the summary. The *diversity* property has attracted considerable attention in query result diversification problem [6, 18, 33], as a means of counteracting the over-specialization problem, i.e. the retrieval of too homogeneous results in recommender systems and web search. The *coverage* is a property widely used in various contexts [4, 20, 22, 32] to represent the number of topics, terms, or information nuggets covered by the summary. It is worth to notice that *diversity* is a property of the summary itself but not related to objects not in the summary, while *coverage* is a property regarding both the summary and the whole data set. These different properties are suitable for different settings. Our goal is to give informative summaries about the whole set which maximizes the recipient's knowledge about the set. We therefore focus on how to model the *coverage* property in this paper.

Comparison to our work. The proposed framework is developed for uncertain multi-attribute data and is capable of factoring correlation. We are unaware of any existing work on extractive summarization that would deal with uncertain attributes – even over a single attribute and without correlation. In addition, most of these existing techniques are developed to summarize over a single deterministic attribute and they do not consider correlation across attribute data values.

8. CONCLUSIONS

In this paper, we have considered the problem of summarizing a set of data objects with multiple uncertain attributes. The goal is to generate a smaller subset (summary) to cover the information contained in the larger set. We have proposed a general summarization framework that models the information contained in objects in a multi-attribute space and optimizes a probabilistic coverage property of the summary. We have also provided a greedy algorithm to the NP-Hard problem of summarization and two efficiency optimization techniques to make the greedy algorithm very efficient and scale very well against dataset size and summary size. Empirical comparisons between our summarization framework and baseline techniques show a clear advantage of our approach.

9. REFERENCES

[1] http://www.alchemyapi.com/.

[2] http://en.wikipedia.org/wiki/Kullback-Leibler_divergence.

[3] http://www.ics.uci.edu/∼dvk/pub/sigmod14ext.pdf.

[4] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. *WSDM'09*.

[5] D. Burdick, Calimlim, and Gehrke. Mafia: a maximal frequent itemset algorithm for transactional databases. *ICDE'01*.

[6] Z. Chen and T. Li. Addressing diverse user preferences in sql-query-result navigation. *SIGMOD'07*.

[7] C. Chui, B. Kao, and E. Hung. Mining frequent itemsets from uncertain data. *PAKDD'07*.

[8] L. Duan and W. N. Street. Finding maximal fully-correlated itemsets in large databases. *ICDM'09*.

[9] K. Ganesan and C. Zhai. Opinion-based entity ranking. *Information Retrieval'11*.

[10] M. Hu and B. Liu. Mining and summarizing customer reviews. *KDD'04*.

[11] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. *MIR'08*.

[12] M. J. Huiskes, B. Thomee, and M. S. Lew. New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. *MIR'10*.

[13] D. Inouye and J. K. Kalita. Comparing twitter summarization algorithms for multiple post summaries. *SocialCom'11*.

[14] D. V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM TODS*, 31(2):716–767, June 2006.

[15] D. V. Kalashnikov, S. Mehrotra, J. Xu, and N. Venkatasubramanian. A semantics-based approach for speech annotation of images. *TKDE'11*.

[16] L. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. *WWW'08*.

[17] S. Kim, M. Barsky, and J. Han. Efficient mining of top correlated patterns based on null-invariant measures. *ECML PKDD'11*.

[18] G. Koutrika, A. Simitsis, and Y. Ioannidis. Precis: The essence of a query answer. *ICDE'07*.

[19] J. Li and A. Deshpande. Consensus answers for queries over probabilistic databases. *PODS'09*.

[20] L. Li, K. Zhou, G. Xue, H. Zha, and Y. Yu. Enhancing diversity, coverage and balance for summarization through structure learning. *WWW'09*.

[21] C. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. *NAACL'03*.

[22] K. Liu, E. Terzi, and T. Grandison. Highlighting diverse concepts in documents. *SDM'09*.

[23] A. Loui and et al. Kodak's consumer video benchmark data set: concept definition and annotation. *MIR'07*.

[24] D. K. Ly, K. Sugiyama, Z. Lin, and M. Kan. Product review summarization based on facet identification and sentence clustering. *CoRR'11*.

[25] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming, 14:265-294, 1978.*

[26] T. Nguyen and H. Lauw. Using micro-reviews to select an efficient set of reviews. *CIKM'13*.

[27] R. Nuray-Turan, D. V. Kalashnikov, and S. Mehrotra. Adaptive connection strength models for relationship-based entity resolution. *ACM JDIQ*, 4(2), 2013.

[28] R. Nuray-Turan, D. V. Kalashnikov, S. Mehrotra, and Y. Yu. Attribute and object selection queries on objects with probabilistic attributes. *ACM TODS*, 37(1), Feb. 2012.

[29] A. E. Savakis, S. P. Etz, and A. C. Loui. Evaluation of image appeal in consumer photography. *ICWSM'10*.

[30] L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: Continuous summarization of evolving tweet streams. *SIGIR'13*.

[31] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. *ICCV'07*.

[32] P. Sinha, S. Mehrotra, and R. Jain. Effective summarization of large collections of personal photos. *WWW'11*.

[33] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. *ICDE'08*.

[34] T. Wu, Y. Chen, and J. Han. Re-examination of interestingness measures in pattern mining: a unified framework. *Data Min. Knowl. Discov., 2010*.

[35] X. Yang, C. M. Procopiuc, and D. Srivastava. Summarizing relational databases. *VLDB'09*.