Credibility-enhanced Curated Database: Improving the Value of Curated Databases

Qun Ni 1, Elisa Bertino 2

1Computer Science, Purdue University
West Lafayette, IN 47907, USA
ni@cs.purdue.edu
2bertino@cs.purdue.edu

Abstract—In curated databases, annotations may contain different opinions from that in sources. Moreover, annotations may contradict each other and contain uncertainty. Such situations result in a nature question: “Which opinion is the most likely correct?” In this paper, we define a credibility-enhanced curated database and propose an efficient method to accurately evaluate the correctness of sources and annotations in curated databases.

I. INTRODUCTION

Curated databases are databases that are populated and updated with a great deal of human effort. There is a long list of applications of curated databases. Most reference works that one traditionally would find on the reference shelves of libraries, e.g. dictionaries, encyclopedias [1], are now curated databases. Many scientific databases, including databases of genetic sequence functions [2], protein functions [3], astronomical phenomena [4], and biodiversity mapping [5], are curated databases too.

The value of curated databases lies in the quality of the data they contain [6]. Different contributors of a curated database may have different opinions on some pieces of existing information or may disagree on the information that should be stored [7]. Relational DBMSs, though they can help database users store, query and update these shared data, provide little support for them to manage conflicting opinions on the correctness of the stored data.

Annotated databases [8], [9] have been recently introduced to address such a drawback. Annotations are meta information that helps explain, correct, or refute some pieces of information in curated databases, referred to as source information, and thus may help improve the quality of curated databases. Annotations, though they have been recognized as a necessary feature for a new generation of DBMSs [6], may contain different opinions on other annotations and source information as well. Therefore, annotated databases are able to help users store and query conflicting opinions [7] but provide little support for them to make a rational judgment on the correctness of source information and annotations.

In other words, annotations are able to help answer questions about the existence of conflicting opinions or the content of these opinions. However, annotations are not able to directly help answer questions such as which opinion (or annotation) is the best or has the highest probability to be the best? The value of a curated database, even if its source information or annotations contain the best opinion, is reduced if the database is not able to provide a reasonable way to help users locate the opinion. Therefore, the capability to help locate the best opinion should be considered a necessary criterion for evaluating the value or quality of a curated database.

In order to answer the question, we need to evaluate the correctness of source information and relevant annotations. In the process to evaluate the correctness of source information and annotations, there are several challenges. First, in the source information or annotations, there may be some uncertainties in opinions [10], i.e. different subjective confidence degrees on different opinions. To determine the correctness of different opinions, we have to somehow quantify these subjective confidence degrees. Second, the credibility or trustworthiness of the contributors of the source information or annotations is crucial but we usually have little reliable and explicit information about them. Third, different opinions, e.g., functions of proteins, taxonomy of animals, classification of stars, usually constitute an ontology. The credibility of contributors in different fields of the ontology usually is quite different. When evaluating the correctness of different opinions and the credibility of a contributor, we need to consider the contributor’s different skills in different fields.

To address these challenges, in addition to the concept of source information and annotations, we introduce the concept of the credibility of contributors in curated databases. A credibility database stores and maintains the credibility of all contributors and thus helps database users to determine the correctness of different opinions and locate the best opinion. The credibility of a contributor is established purely based on the contributor’s historical quality records or correctness rates on source information and annotations. Meanwhile, the correctness of source information and annotations is evaluated based on the credibility of relevant contributors. Specific measures on relevant factors are proposed for computing the correctness of the source information and annotations and for computing the credibility of contributors. To summarize, we make the following contributions in this paper:

- We define a formal data model for credibility-enhanced curated databases and establish the connection between credibility, annotations, and source information in a curated database based on a real application (Section II-III).
- We design measures to compute the credibility of contributors and the correctness of source information on relevant factors, including ontology, and a propagation algorithm to...
evaluate them (Section IV).

- We prove that the propagation algorithm converges in finite number of steps if a simple condition is satisfied. We show that the condition is easily satisfied in practice and in a simulation (Section IV - V).
- We show that the propagation algorithm is highly effective and efficient through a carefully established simulation. The algorithm usually converges within six loops regardless of the number of contributors and annotations and is able to significantly improve the quality of curated databases (Section V).
- We investigate the problem of the quantification of subjective confidence degrees on opinions through experiments and obtain a surprising but interesting result. A quantification solution based on randomly generated numbers usually provides better results than another solution based on some rules (Section V).
- We discuss the dynamics of credibility and the problem of the computational complexity for very large scale curated databases and propose a practical solution (Section VI).

Last but not least, we would like to emphasize that the application of our approach is well beyond scientific databases. Finding the best opinion or the opinion with the highest possibility of being the best from a collection of uncertain subjective opinions, whose sources have unknown credibility, is not only quite common in practice but also very important in some situations, e.g. in a decision making case. Our propagation algorithm is able to address this problem and improve the quality of decisions based on these opinions. For instance, in our daily life, there are plenty of review systems, such as the feedback system on buyers and sellers (eBay.com), the review system on merchandise (Newegg.com and Amazon.com), and the user rating system on movies (imdb.com) and resellers (resellerratings.com). They share similar properties, e.g. subjective confidence degrees on reviews, questionable credibility of reviewers, and ontology (e.g. the categories of merchandise and movies). All these systems adopt a simple average process to compute the final review score. Our propagation algorithm is able to more accurately evaluate the correctness of these review scores and generate a much better final score. Therefore, the adoption of our approach in these scenarios may effectively improve the quality of our lives by saving money, time or efforts on enjoying better products.

II. Motivating Application

The NatureMapping program [11] that motivates this paper allows citizens of all ages conducting meaningful science for the benefit of their local communities and biodiversity by asking the public to report “What do you see and where do you see it?” and create a national public biodiversity database for all to use. The NatureMapping program requires the following steps.

**Step 1 Source (raw) data collection.** Volunteers who have taken a training workshop for NatureMapping collect raw data using the form shown in Fig. 1. Some fields are easily understood. Other fields are explained as follows. Species name records the common name of an animal. Species code contains four or five letters abbreviated from the scientific name of the animal. The first two letters of the code are the first two letters of the genus name. The second two or three letters of the code are the first two or three letters of the species name. For instance, the scientific name of anna’s hummingbird is Calypte ANna, its species code is thus CAAN. Unsue is used for questionable sighting. Three options “1”, “2”, or blank can be put in the cell. “1” means that you are not sure if you identified the species correctly. “2” means that you know the species ID is correct, but it “shouldn’t be there”. Blank means that you are pretty sure about your observation. “Yes” in Estimate indicates that how many? is an estimation of the number of animals in a large flock. Habitat code is a three digit code that describes the space an animal is using, e.g., 236 indicates a developed (2), light (3), wooded forests (6) habitat. The probable value for the how observed field are (1) museum - many records originally came from museum collections, (2) heard, (3) saw, (4) trapped, (5) sign (tracks, scat, pellets, fur or hair, feathers, bones), (6) heard and saw, (7) photos, (8) on nest, (9) dead, (10) other database, (11) flying overhead. Please note, the observers are recommended including photos or specifying field notes (not shown in Fig. 1) that sketch and describe marks on head, body, and tail for mammals, birds, amphibians, and reptiles, e.g. color, behavior, body covering, size, temperature, and weather conditions.

![Data Collection Form](image)

Volunteers have to transform the source data into a standard nature mapping database format (See Fig. 2) in which red attributes are mandatory before submitting the source data to the NatureMapping program. Please note that blank values in mandatory attributes have specific meaning given in the explanation of raw data fields, e.g., blank value in “estimate” means that the quantity (QTY) is not an estimation. The “Q” and “comment” attributes correspond to “Unsure” and “Field notes” in the raw data, respectively. Secondary habitat code (Hab_2) is used when the location is around water. First habitat code identifies the water habitat and Hab_2 code describes the adjoining habitat.

**Step 2 Annotation.** Source data submitted by volunteers may contains errors, and thus will be curated by experienced volunteers or experts before being published in biodiversity maps. There are several sources of errors. First, the animal is not sighted or heard clearly, or the sign does not contain sufficient information. Second, the observer does not have sufficient knowledge for judging the animal or judges the animal by a mistake. In practice, it is quite common that two team members have different opinion...
on an animal [7]. Even experts can make a mistake easily for some animals with similar appearances. Therefore, a peer review process is adopted to improve the quality of biodiversity maps. A review, referred to as an annotation, may reference to a source record or another annotation record. One peer review can be based on an source record or another annotation.

\[ c(\vec{x}, \vec{y}) = \sum_{i} x_i y_i \]

where \( x_i \) represents the confidence degree of an opinion from observer \( i \) and \( y_i \) represents the credibility of observer \( i \) to integrate different confidence degrees of the specific opinion, we obtain the following confidence degree from all annotations for each opinion.

\[ c_{\text{anna's}} = \frac{(1 \times 0.2 + 0.5 \times 0.3)}{(0.2 + 0.3)} = 0.7 \]

\[ c_{\text{allen's}} = \frac{(0.75 \times 0.1 + 0 \times 0.4)}{(0.1 + 0.4)} = 0.15 \]

Now we can safely draw a conclusion that the animal is an Anna’s hummingbird because the final confidence degree to be an Anna’s hummingbird is 0.7 and the final confidence degree to be an Allen’s hummingbird is 0.15.

To apply this approach, there are two challenges. First, how to obtain the credibility of each observer. To evaluate the credibility, we propose different measures based on the properties of the problem and develop corresponding algorithms in Section IV. Second, the quantification of confidence degree is just educated guess, therefore, how to justify the quantification of subjective confidence degrees. This topic will be addressed in Section V-D.

### III. CREDIBILITY-ENHANCED CURATED DATABASE

As mentioned in the introduction, the approach we propose can be used to solve a series of similar problems. Before presenting our approach, we formalize these problems by introducing some basic notions and definitions of a credibility-enhanced curated database, referred to as a CC database, which extends curated databases with an additional attribute to maintain confidence information and additional relations to maintain credibility and ontology. Based on these formalization, we thus are able to propose a general approach that can deal with a family of problems.

For simplicity, we apply relational database concepts to define a CC database that contains a source relation, an annotation relation, a contributor relation, and an ontology (see Fig. 4).

**Definition 1 (Source):** A source relation \( S \) is a set of tuples (\( sid, cid, concept, confidence, datatype, op_1, \ldots, op_n \)) where \( sid \) is the primary key; \( cid \), a foreign key, references to the primary key in a contributor relation; \( concept \) is the opinion from contributor \( cid \); \( confidence \) represents the confidence degree from contributor \( cid \); \( datatype \) represents the time stamp; \( op_1 \ldots op_n \) represents other optional attributes.

An example of a source relation is the NaturalMapping database shown in Fig. 2. An annotation relation shares some attributes with the source relation.

**Definition 2 (Annotation):** An annotation relation \( A \) is a set of tuples (\( aid, sid, rid, cid, concept, confidence, datatype, op_1, \ldots, op_n \)) where \( aid \) is the primary key; \( sid \), a foreign key, references to the primary key in a source relation; \( rid \) references
to another primary key in the annotation relation; other fields are the same as those in Definition 1.

An example of the annotation relation is the annotation in Fig. 3 where oid corresponds to cid in the definition.

**Definition 3 (Contributor):** A contributor relation C is a set of tuples (cid, op1, ..., opn) where cid is the primary key; op1...opn represent other optional attributes related to a contributor.

A contributor relation records the location and communication information of a contributor (as shown in the title of Fig. 1). For instance, in the NaturalMapping program, every observer should register her personal information before submitting her observation that is stored in the contributor relation [12].

Even if credibility is relevant to contributors, it cannot be stored together with contributors because credibility is relevant to concepts or their categories as well. Moreover, credibility may be dynamic and vary as time goes by. We, thus, introduce the following credibility relation.

**Definition 4 (Credibility):** A credibility relation CR is a set of tuples (cid, fromdate, todate, concept, credit) where cid, fromdate, todate, and concept are the multiple primary keys; credit represents the corresponding credibility degree.

An example of a credibility relation is shown in Fig 5. The NaturalMapping program has been running since 1992, and some long-standing members have significantly improved their capabilities to correctly discern species that are hard to distinguish. Such a phenomenon indicates that an observer has different credibility in different time periods, just like FICO credit scores. In order to evaluate the confidence degree of some annotations recorded in 2006, we should apply credibility in 2006 other than that in 2009. Therefore, we introduce a temporal field for the credibility of each contributor. Another observation is that each contributor usually has specific expertise on some species and may be less capable of another species. The observation motivates the concept field in the credibility relation. We will elaborate this topic in Section IV-G.

Concepts usually have their taxonomy; we thus introduce an ontology relation to maintain the taxonomy.

**Definition 5 (Ontology):** An ontology relation On is a set of tuples (subject, object, predicate) where subject and object are the multiple primary keys; predicate represents the relation between a subject and an object.

Assuming a predicate to be “is a parent of”, the semantics of an ontology tuple is “the subject is a parent of the object”. Such a definition is based on a well developed ontology language Resource Description Framework (RDF) from Semantic Web [13]. The ontology is used to specify the relation between concepts, e.g. the scientific classification of species as shown in Fig. 6, hummingbirds is a parent of the anna’s hummingbird. The concepts applied in source and annotation relations are from the ontology relation.

**IV. BUILDING CREDIBILITY**

As mentioned earlier, one challenge in computing the correctness of each concept is the lack of credibility information for each contributor. The credibility of a person is built by her personal behavior records. Likewise, we may build the credibility of each contributor based on the contributor’s historical records. Intuitively, if a contributor’s annotation is correct, e.g. to be chosen as the final species in the NaturalMapping program, the contributor gains credibility. By contrast, if a contributor’s annotation is wrong or not so precise, the contributor loses credibility. Based on this observation, we develop our propagation algorithm to calculate the credibility of each contributor.
Section IV-F.

A. Propagation

Similar to the expectation-maximization algorithm [14], the propagation algorithm contains two iterative steps:

1) Judgment stage: it determines the “best” concepts for each set of a source record and relevant annotation records. The algorithm calculates the correctness degree of each record in the set by the product of the confidence degree of the record and the credibility of its contributors. All correctness degrees will be summed separately for each concept in the set. The concept with the highest correctness degree is chosen to be the “best” concept in the stage.

2) Credibility stage: it computes the credibility of each contributor based on the the contributor’s performance on providing “correct” concepts (gain credibility), which is the same concept as the “best” concept in the previous judgment stage, in the contributor’s source records and annotation records, and the contributor’s performance on providing “wrong” concepts (loss credibility).

Steps 1 and 2 are repeated until the credibility converges.

Assume that we have 3 contributors X, Y, and Z, two source records S1 (by X) and S2 (by Y), and two annotation records A1 (by Z, for S1) and A2 (by Z, for S2). From X’s perspective, X’s credibility propagates to Z in the first loop. Due to the connection between S1 and A1, Z’s credibility is affected by X’s in the first loop. Thus Z’s credibility contains factors from X’s.

Even if Z’s credibility contains factors from X’s, it does not mean that Z’s credibility will increase. Whether Z’s credibility increases, decreases, or does not change depends on the correctness of Z’s concept and that of X’s concept.

In the second loop, from Z’s perspective, Z’s credibility affects Y’s due the connection between S2 and A2. Since before the second loop Z’s credibility has already contained factors from X’s, Y’s credibility now contains factors from X’s, i.e., X’s credibility propagates to Y’s.

Likewise, from Z’s perspective, Z’s credibility propagates to X’s and Y’s in the first loop. From Y’s perspective, Y’s credibility propagates to Z’s in the first loop and to X’s in the second loop.

The key idea in the algorithm is propagating the credibility from each contributor to other contributors who directly or indirectly connect to the contributor until the credibility of each contributor reaches a stable state in the credibility stage such that the change of the credibility of contributors between this loop and that in the previous loop is negligible. It is the reason why we call it “propagation” algorithm.

Experts usually have higher credibility than non-professionals. Through the propagation algorithm, the expert’s and non-professional’s credibility will finally affect each other 1. However, if we intentionally put more weights on expert’s credibility and smartly choose measures of credibility, the propagation algorithm has the potential to reach a final stage with a more accurate credibility ranking for all contributors, including contributors who are not directly connected to any experts. In what follows, we discuss various measures used in the algorithm.

B. Correctness of Concepts

Given a source record, there may be n annotation records. All these n + 1 records may contain m different concepts where m ≤ n + 1. Assuming concept x is one of m concepts, the correctness of concept x is calculated by the following measure

\[
\text{correctness}_x = \sum_{y=1}^{l} \text{confidence}_y \times \text{credit}_z
\]

where l represents the number of records that contain concept x, confidence_y represents the confidence degree of record y (one of the l records) and credit_z represents the credibility of contributor z who authored record y.

Finally, we obtain m different correctness values. The concept with the highest correctness is chosen as the “correct” concept for these n + 1 records in this loop.

C. Raw Credibility

The credibility calculation is more complicated. First, assuming that all concepts belong to n top categories, e.g., birds and mammals, we may want to measure the credibility of each contributor differently for each category because each contributor may have expertise only on some categories. Second, there are different notions of credibility in this process. The first one is raw credibility that counts the difference between the number of correct concepts and the number of wrong concepts provided for each contributor.

In order to compute raw credibility for each contributor, we need to first accumulate raw credibility, stored in variable “sum of raw credibility”, from each records curated by the contributor.

If a record r from a contributor x, either a source record or annotation record, contains the “correct” concept y in top category z identified in the previous step, we will accumulate the sum of raw credibility of the contributor.

\[
\text{sum of raw credit}_{x}[z] = \text{sum of raw credit}_{x}[z] + \text{Confidence}_r \times 1
\]

The intuition underlying the notion of raw credibility is that if contributor x provides a correct opinion, we should increase the contributor’s credibility of one unit weighted by her confidence degree in the concept’s category.

Intuitively, if a record r from a contributor x, either a source record or annotation record, contains the “wrong” concept y in top category z identified in the previous step, we should punish the contributor’s credibility on category z.

\[
\text{sum of raw credit}_{x}[z] = \text{sum of raw credit}_{x}[z] - \text{Confidence}_r \times 1
\]

Moreover, assuming the “correct” concept’s category is w, that is, different from z, the “wrong” case shows that the contributor

\[1\] There is a precondition for this situation to happen, please reference to Section IV-F.
cannot recognize a concept in category \( w \). Therefore, we should punish the contributor’s credibility on category \( w \) as well.

\[
\text{sum of raw credit}_{x}[w] = \text{sum of raw credit}_{x}[w] - \text{Confidence}_{r} \times 1
\]

To summarize, for a “correct” case, we only increase the contributor’s credibility in one category by one weighted unit. However, for a “wrong” case, we may decrease the contributor’s credibility by one weighted unit in two categories.

After processing all records, we may obtain the raw credibility of contributor \( x \) on category \( z \).

\[
\text{raw credibility}_{x}[z] = \frac{\text{max}(\text{sum of raw credit}_{x}[z], 0)}{n}
\]

where \( n \) represents the number of records by \( x \) that contain concepts in category \( z \). The max function is used to ensure that the raw credibility cannot be a negative number, i.e. in the unit interval \([0, 1] \). Such a restriction is important for our propagation algorithm to converge.

The raw credibility cannot be used in the calculation of the correctness of concepts but it is the base of other credibility calculations. There are two other notions of credibility, namely normalized credibility and source credibility, that are discussed in the next section.

D. Normalized Credibility and Source Credibility

As stated in the beginning of this section, the basic idea of propagation is propagating the “credibility” from professional contributors (experts) to non-professional volunteers. Based on the credibility of professionals, we could gradually build up the credibility of volunteers. In practice, we have these experts served in almost every curated database. For instance, in the NaturalMapping program, some biologists review the submissions from volunteers. It is useful for us to choose opinions from some professional experts, referred as to the trusted sources, as “standards” to evaluate the credibility of other contributors.

Obviously the credibility of these trusted sources should be higher than other contributors, otherwise, the judgments from trusted sources can be easily subverted by non-professionals whose number is usually much larger. To realize this objective, we apply the following three steps to generate the final credibility of contributors from their raw credibility.

1) Normalization of raw credibility. We normalize the credibility based on the number of contributors and the category.

\[
\text{normalized credibility}_{x}[z] = \frac{\text{raw credibility}_{x}[z]}{\sum \text{raw credibility}_{y}[z]}
\]

After the normalization, we weaken the opinion from each contributor based on the total number of contributors while maintaining the order of their credibility.

2) Generation of source credibility. We assign the source credibility to trusted sources based on the following formula (assuming \( n \) to be the number of trusted sources).

\[
\text{source credibility}_{x}[z] = \begin{cases} 
1/m & i: \text{trusted source} \\
0 & i: \text{non-trusted source}
\end{cases}
\]

3) Integration of normalized raw credibility and source credibility. Let \( \alpha \) be a real number in \([0,1] \), \( \text{rcredit}_{x}[z] \) be the vector of normalized raw credibility on category \( x \), and \( \text{scredit}_{x}[z] \) be the vector of source credibility on category \( x \), we calculate the vector of credibility \( \text{ccredit}_{x}[z] \) for each contributor based on the following formula.

\[
\text{ccredit}_{x}[z] = \alpha \text{rcredit}_{x}[z] + (1 - \alpha)\text{scredit}_{x}[z]
\]

The choice of \( \alpha \) depends on applications (See Section V-E for details).

E. Initial Credibility

We have to set an initial credibility for each contributor, otherwise, we cannot start the propagation algorithm. Assuming we have \( n \) contributors, we set \( 1/n \) to be the initial credibility for each contributor on each category.

F. Connectivity

In practice, it is possible that no experts are connected with a small group of contributors. This situation happens if contributors in such a small group only comments on each other’s records and no contributors from other groups are involved in the group’s activities. Such a situation not only may hurt the final accuracy due to the lack of expert’s opinions but also may result in our propagation algorithm not being able to converge. To address this issue, we introduce the following concept.

Definition 6 (Connectivity): Let \( x, y \) be two contributors in contributor set \( C \), a binary relation \( c(x,y) \) on \( C \) is called a connectivity relation if \( x \)’s credibility can affect \( y \)’s credibility in finite steps through the propagation algorithm.

It is easy to prove the following lemma based on Definition 6.

Lemma 1: The connectivity relation \( c \) is an equivalence relation on the set of contributors, i.e. the following properties hold.

- \( c(x, x) \)
- if \( c(x, y) \) then \( c(y, x) \)
- if \( c(x, y) \) and \( c(y, z) \) then \( c(x, z) \)

Since the connectivity relation is an equivalence relation, we obtain the following definition.

Definition 7 (Equivalence Class): Let \( x \) be a contributor in contributor set \( C \), the equivalence class of \( x \) under connectivity relation \( c \), denoted \( [x] \), is defined as \([x] = \{ y \in C | c(x,y) \} \).

Based on the Definition 7 we can easily prove the following lemma.

Lemma 2: Given an contributor \( x \) in contributor set \( C \), \( x \) connects all other contributors iff \( C \) is an equivalence class.

Now it becomes clear that we need to verify whether the contributor set is an equivalence class in order to guarantee an expert’s opinion to affect all contributor’s credibility and ensure that the propagation algorithm converges. An algorithm is proposed to calculate the equivalence classes in contribution records based on source records and annotation records. The crucial observation is that the contributor of each source record and the contributors of relevant annotation records constitute an equivalence class (a set). We, thus, can pair-wisely union these sets if the intersection of these sets is not empty until
the interaction of any two sets left is empty or only one set is left. The latter case confirms that the contributor records form an equivalence class. If the former case happens, it is easy to connect these sets in practice by asking some random contributors from the largest set to annotate any source records or annotation records authored by contributors in other sets.

The algorithm to verify whether a contributor relation is an equivalence is shown in Algorithm 1. Line 2-7 create the list of equivalence classes based on each source record. Line 9-19 pair-wisely union equivalence classes whose intersection is not empty. The complexity of line 9-19 dominates that of line 2-7. Assuming \( n \) to be the number of source records, the loop runs in \( O(n) \) in its worst case. All set operations run in \( O(m) \) assuming \( m \) to be the number of contributor records. Therefore, the complexity of Algorithm 1 is \( O(n^3m) \).

### Algorithm 1: DetectEQClass

**Input:** sr: Source Relation; ar: Annotation Relation; cr: Contributor Relation.  
**Output:** ec: List of Equivalence Classes

```java
Algorithm 1: DetectEQClass(sr, ar, cr)
Input: sr: Source Relation; ar: Annotation Relation; cr: Contributor Relation.
Output: ec: List of Equivalence Classes
1 List ec = new List();
2 foreach Source s in sr.row do
3   HashSet hs = new HashSet();
4   hs.Add(cr.row[s.cid]);
5   foreach Annotation a in ar.row do
6     if s.sid == a.sid then hs.Add(cr.row[a.cid]);
7     ec.Add(hs);
8 int lastNumOfElement = 0;
9 repeat
10   lastNumOfElement = ec.Count;
11   for (i = 0; i < ec.Count; i++) do
12     for (j = i + 1; j < ec.Count; j++) do
13       HashSet temp = new HashSet(ec[i]);
14       temp.IntersectWith(ec[j]);
15       if temp.count != 0 then
16         if s.sid == a.sid then hs.Add(cr.row[a.cid]);
17         ec.Add(hs);
18         j = j - 1;
19       until ec.Count == lastNumOfElement;
20 return ec;
```

### G. Granularity of Credibility

We have already adopted top level concepts (precisely level 2) to divide credibility into different sets. However, there are different choices on the level of concepts in the ontology to be used to divide credibility into different sets. The level controls the granularity of credibility on concepts. Assuming the height of the ontology to be \( n \), if we choose the level 1 concept, e.g. animal in the Fig. 6, the credibility is such coarse-grained that we may mix the low credibility of a contributor on birds with her high credibility on fishes. If we choose level \( n \) concept, the credibility is such fine-grained that we may not apply the credibility on a concept to another similar concept in a same family. An appropriate choice of a concept level depends on applications, e.g., level 2 concepts (birds, fish, mammals) seems to be a good choice in the NaturalMapping program.

### H. The algorithm

We present the propagation algorithm in this subsection. As shown in Algorithm 2, Lines 1-2 initialize the credibility of trusted sources and the normalized credibility. Line 6 computes the credibility used in every loop. Line 7 clears raw credibility that may contain values in the last loop. Lines 9 - 10 collect all concepts in a source record. Line 11 computes the correct concept based on our correctness measure. Lines 12-14 compute the sum of raw credibility using our sum of raw credibility measure. Line 15 computes the raw credibility. Line 16 computes the normalized credibility. Lines 18-20 computes the sum of difference between credibility in this loop and credibility in the last loop.

### Algorithm 2: Propagation

**Input:** sr: source relation; ar: annotation relation; cr: contributor relation  
**Data:** credit: credibility; sCredit: source credibility; rCredit: raw credibility; nCredit: normalized credibility; \( \delta \): convergence threshold; cSet: the list of concepts in a source record and relevant annotation records; cConcept: the correct concept for each source record

```java
Algorithm 2: Propagation(sr, ar, cr)
Data: credit: credibility; sCredit: source credibility; rCredit: raw credibility; nCredit: normalized credibility; \( \delta \): convergence threshold; cSet: the list of concepts in a source record and relevant annotation records; cConcept: the correct concept for each source record
1 Initialize sCredit for each expert on its special category to be \( 1/n \) where \( n \) is the number of experts in the category and for other cases to be 0;
2 Initialize nCredit to be \( 1/m \) where \( m \) to be the number of contributors;
3 sum = 0;
4 repeat
5 lastsum = sum;
6 credit ← \alpha sCredit + (1 - \alpha) \times nCredit;
7 rCredit.Reset();
8 foreach s in sr do
9   cSet ← s;
10  foreach Annotation a in ar do cSet ← a;
11  cConcept ← CorrectConcept(cSet, credit);
12  sumRawCredit ← AccumulateRawCredit(s, cConcept);
13  foreach a in ar do
14    sumRawCredit ← AccumulateRawCredit(s, cConcept);
15    rCredit ← sumRawCredit.value / sumRawCredit.number;
16    nCredit ← rCredit.value / rCredit.sumOfValue;
17    sum = 0;
18  foreach c in cr do
19    sum = sum + \alpha nCredit[c] + \alpha sCredit[c] - credit[c];
20    nCredit[c] = (1 - \alpha)\times nCredit[c] + \alpha sCredit[c];
21 until |sum - lastsum| < \delta ;
```

The complexity of the propagation algorithm depends on how quickly the credibility will converge. We show that the algorithm
converges very quickly in practice by experiments (Section V).

**Theorem 1 (Convergence):** If a contributor set is an equivalence class, algorithm 2 converges in a finite number of steps on the set.

**Sketch:** We can translate the calculation in the credibility stage of our propagation algorithm into an equivalent calculation on review scores from one contributor to another contributor based on the agreement or disagreement on relevant concepts. Such a review score calculation forms a Markov chain whose transition matrix is irreducible (equivalence class) and aperiodic (the $\alpha$ factor), and contains finite elements (contributors). Therefore, the calculation must converge in finite steps.

**V. Experiments**

The NaturalMapping program, unfortunately, does not maintain an annotation database even if there is a review process on observations submitted by volunteers. To evaluate the effectiveness and efficiency of our approach, we had to carry out a simulation. However, synthetic data that is purely randomly generated is meaningless for our purpose. Therefore, based on our observation on the NaturalMapping program and some well known distributions of social activities, we designed a data generator for our experiments.

The first observation is that each contributor has a hidden correctness factor. The factor is represented by the number of source records and annotation records divided by the number of total source records and annotation records from each contributor. Moreover, each contributor has different correctness factors on different high level concepts, e.g. birds or mammals.

The number of source records submitted by contributors follows the well-known power-law distribution, or the “80-20” rule. Many social activities follow the power-law distribution, e.g. the eBay feedback score system [15], in that most contributors only submit few sources and annotations but a small portion of contributors will submit most of sources and annotations. To be precise, we adopt a Pareto distribution

$$P(x) = \frac{k \cdot x^k}{x^{k+1}}$$

with a Pareto index $k = \log_2^5$ (exactly an “80-20” distribution).

A very small portion of contributors are experts. Experts tend to contribute a slightly higher number of source records than volunteers do. Therefore, we choose experts from contributors who are above average. Experts have higher correctness factors, especially in their area of expertise.

The time of each source record follows the well-know Poisson process because each observation is a small probability event and there is usually no or little dependence between two different observations. Many small probability social activities, e.g. telephone calls, can be described by a Poisson process

$$P[(N(t + \tau) - N(t)) = k] = \frac{e^{-\lambda \tau}(\lambda \tau)^k}{k!} \quad k = 0, 1, \ldots$$

where $N(t + \tau) + N(t)$ is the number of observations in time interval $(t, t + \tau)$, $\lambda$ is the average rates of observations. The observation dates are generated based the Poisson process.

Given each source record in the NaturalMapping program, there is only one truth (or concept) relevant to the animal. However, the probability that the truth is discovered by an observer depends on her correctness factor. Both the truth and the concept determined by the observer are recorded in the source records. The truth is used for the verification purpose in our experiment.

One or more annotation records are generated for each source record. Contributors who have expertise in the source record more likely comment on the record. Experts on the source record have slightly higher probability annotate the source record.

If there are two or more equivalent classes in the data generated, one random member in the largest class is picked up to annotate one source record in different classes to connect different equivalent classes.

All aforementioned factors can be adjusted by parameters. Due to the space limitation, we only list here the values of the most important parameters adopted in our experiment:

- The ratio between contributors and source records is 1:4.
- The ratio between source records and annotation records is 1:3.
- The probability of experts in contributors that is above average is 0.1. Thus the total percentage of experts is below 5%.
- The expert correctness factor follows in the domain $[0.9, 0.99]$ (skilled) and $[0.5, 0.9]$ (non-skilled).
- The contributor correctness factor follows in the domain $[0.4, 0.9]$ (skilled) and $[0.1, 0.4]$ (non-skilled).
- The probability that contributors skilled in a certain concept annotate a source record is 0.6.
- The probability that experts skilled in a certain concept annotate a source record is 0.2.

The ontology adopted in the experiments is the one from the NatureMapping program that contains 680 species in five top level categories, fish, mammals, birds, reptiles, and amphibians.

The first thing to assess is whether contributors in the generated data sets are equivalence classes. We obtain 100 data sets from the generator based on different parameters, and the condition is always satisfied. We cannot say that the condition can be always satisfied in practice, but we may safely say that the condition may be easily satisfied in practice.

Five data sets based on different numbers of contributors are generated as shown in Fig. 7. In the following subsections, we analyze the effectiveness and efficiency of our propagation algorithm based on these data sets.

<table>
<thead>
<tr>
<th>Number of contributors</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>500</td>
<td>1000</td>
<td>2000</td>
<td>4000</td>
<td>8000</td>
</tr>
<tr>
<td>Number of sources</td>
<td>1850</td>
<td>5371</td>
<td>8955</td>
<td>18641</td>
<td>42579</td>
</tr>
<tr>
<td>Number of annotations</td>
<td>5568</td>
<td>16145</td>
<td>26661</td>
<td>55990</td>
<td>127893</td>
</tr>
</tbody>
</table>

**Fig. 7.** Data sets
A. Effectiveness

Fig. 8 displays the number of the wrongly recognized concepts in each data set and relevant accuracy before and after running our propagation algorithm. Because we set the correctness of non-skilled contributors to be [0.1, 0.4] and the non-skilled contributors have the highest probability to submit source records, the accuracy of original source records is only 40%. As we can see from Fig. 8, the accuracy increases from below 40% to 80%. Notice that the proportion of records from experts is very small, the 80% overall accuracy proves the effectiveness of our algorithms.

![Fig. 8. Effectiveness](image)

B. Efficiency

Fig. 9 shows the efficiency of our propagation algorithm. The algorithm converges very fast. As shown in Fig. 9, it usually takes only 5 or 6 judgment and credibility loops to converge regardless of the size of the data set. In the next subsection we show that the 3 loops can achieve sufficiently good accuracy. Further, notice that the data set sizes increase exponentially, the running time is polynomial in the number of contributors, sources records, and annotation records. The running time on the largest data set E with one hundred and seventy thousands records needs 3210 seconds. Data Set E roughly meets the data scale of the NaturalMapping program. However, for much larger curated databases, e.g. the Global Biodiversity Information Facility [5], it would be expected that the running time be of the scale of days. A solution to this problem is discussed in Section VI.

![Fig. 9. Efficiency](image)

C. Convergence

Fig. 10 shows the trends of the converging process based on data set C. Loop numbers are shown in the horizontal axis. The converging processes for all five data sets are similar. The current threshold is set to 0.01. However, as we can see from Fig. 10, we can obtain an almost equally good result by setting the threshold to be 0.1 which requires only 3 loops. From loop three, we could only obtain marginally better results: 1844(3), 1824(4), 1818(5). All five data sets are similar in terms of the converging process.

D. Quantifying Subjective Confidence Degrees

In Section II, we briefly discussed how to quantify the subjective confidence degrees represented by natural languages. In this section, we investigate this issue through experiments.

One key observation is that there is usually a strict order over subjective confidence degrees. In the source records of the NaturalMapping program, there are 3 different confidence degrees, pretty sure (empty), unsure (1, pretty sure but should not be there), unsure (2, not sure). It is reasonable that “empty” > “unsure(1)” > “unsure(2)” and that “pretty sure” > “very likely” > “likely” > “unlikely” > “impossible”.

One solution to quantify these subjective confidence degrees is as follows. First, we set the highest confidence degree to be 1 and the lowest confidence degree to be 0. Second, we equally divide the range between 1 and 0 by the number of other confidence degrees. For instance, we quantify the aforementioned confidence degrees to be (1) pretty sure, 0.75(very likely), 0.5(likely), 0.25(unlikely), 0 (impossible). If we consider that “unsure(2)” is as same as “likely”, we have 1 (empty), 0.75(unsure 1), 0.5(unsure 2). We adopt this method to illustrate the calculation in Section II.

Assuming the number of subjective confidence degrees to be n, the second solution is: (1) randomly generate n different numbers in the real unit interval [0,1]; (2) assign these numbers to n confidence degrees sequentially after sorting them. The motivation of solution two is that we do not know exactly what these numbers for subjective confidence degrees should be and only know their order. Therefore, randomly picking up 5 numbers might not be a bad guess, and repeating this process might help us find the best quantification.

One surprising result, when we evaluate these two solutions, is that after convergence, the second solution always generates slightly better results on the five data sets in terms of accuracy, as shown by the left graph in Fig. 11. Moreover, the required loop number is usually smaller thus resulting in a better converging time. The percentage of the running time of the second solution is shown by the right graph in Fig. 11. Because the results from solution 2 are always better, we believe that such an observation is not a special case. We, thus, run 50 experiments using solution two on data set C to further investigate this issue. The statistics about the results on the

Fig. 10. The converging process on data set C

Language: C#. CPU: Core 2 Duo 3.2GHz, Memory: 3GB, OS: Vista SP2.
number of wrongly identified concepts from these 50 samples after converging are shown in Fig. 12 and its box plot is shown in Fig. 13.

![Box plot of the number of wrong concepts](image)

The number of wrongly identified concepts in data set C using solution one is 1936 that is larger than all results from the 50 samples and is classified as an outlier in the box plot because it is 1.5 IQR (inter-quartile range) higher than the third quartile. There must be some reason for this “counter-intuitive” result. After several tries on different quantifications of subjective confidence degrees, we identify two reasons behind this interesting phenomenon.

- 0 and 1 are too rigid for “impossible” and “pretty sure”, respectively. There is a very low possibility that “impossible” and “pretty sure” are in practice equal 0 and 1, respectively.
- Similarly, the probability that equally distanced confidence degree numbers really happen in practice is very low too.

Based on these two observations, we can imagine, intuitively, that the probability that solution one is a bad guess on subjective confidence degrees is very high. Meanwhile, the probability that solution two is a bad guess is low, as shown in Fig. 13. Therefore, the lesson that we learned here is that if we do not have time to tune the quantification of subjective confidence degrees, the solution two usually gives acceptable results.

We may obtain a sub-optimal quantification of subjective confidence degrees by the following two steps:

1) Randomly picking up a non-trivial sample set from the data set and requiring experts to identify these samples as precisely as possible.

2) Running a Monte Carlo simulation on the sample set to determine its optimal quantification of subjective confidence degrees.

It is highly possible that the optimal quantification of the random set is quite close to the true optimal quantification of the whole data set. However, such a process is very expensive and sometime the accuracy improvement may not be able to compensate its cost.

E. The Weight on Expert’s Opinions

As discussed in Section IV-D, $\alpha$ values determine the weight of expert opinions over other opinions. The higher the $\alpha$ value is, the smaller the loop number of the converging process is, as shown in Fig. 14. If the correctness factor of experts is much higher than that from others, a large $\alpha$ value helps to realize better results, e.g. in our data sets. However, even in this case, overly large $\alpha$ values are still able to hurt the accuracy, e.g., 0.89 or 0.99 in Fig. 14, because it is very hard for others to subvert expert’s wrong opinions.

![The weight on expert’s opinions](image)

As shown in Fig. IV-D, similar to the random choice of the quantification of subjective confidence degrees, our propagation algorithm is not sensitive to the choice of $\alpha$ values. The optimal $\alpha$ values depends on the properties of a data set, for instance, in a situation in which the correctness factor of experts is not much higher than that from others, a small $\alpha$ value may be a better choice than a large one. The expense for finding the optimal $\alpha$ value usually cannot compensate the cost of this process.

F. Comparison

In this last experiment, we compare following cases in terms of the number of wrong concepts (a smaller number is better):

- (i): the fully functional propagation algorithm.
- (ii): the propagation algorithm without considering ontology, i.e. the algorithm does not consider different credibility on different concepts.
- (iii): the propagation algorithm without considering experts ($\alpha = 0$), i.e. experts have the same weights on their credibility as others.
- (iv): the algorithm purely based on experts’ opinions ($\alpha = 1$). There is no propagation. If there are experts’ opinions in either a source record or relevant annotation records, the final concept is roughly based on majority voting. If there are not, the final concept is the concept in the source record.
- (v): the algorithm purely based on the weighted average based on the confidence degree on the source record and relevant annotation records (no expert, no propagation).
The comparison result on data set C is shown in Fig. 15. From this figure, we can see that the fully functional propagation algorithm is significantly better than other cases, which confirms our approach based on the propagation of expert’s opinions.

VI. DISCUSSION

For some curated databases, we may have to calculate the credibility of contributors periodically, i.e. calculate the credibility separately based on different date segments. There are two reasons for this situation. First, some curated databases contain a huge amount of information. Calculating the credibility for each contributor based on the whole data sets may not be practical. Because information in curated databases is gradually accumulated, it is possible for us to divide the information into smaller sections based on different time segments and to calculate the credibility of contributors separately. Second, some curated databases are long period projects, and the credibility of contributors may vary with time. Therefore, the calculation of credibility of contributors should a dynamic process, that is, it should be calculated continuously. Once sufficient new records have been added for a contributor, we need recalculate her credibility.

However, in both cases, since volunteer contributors may continuously improve the quality of their contributions during a non-trivial time period, if we generate the final credibility by treating the credibility from different time period equally, such credibility will hurt the “true” correctness of their recent records.

To mitigate this problem, we introduce an incremental credibility calculation and a discounting factor for calculating the final credibility. The incremental credibility calculation means that we group the records into different sets based on some criteria, e.g. some predefined set cardinality or some predefined time period (half a year). We calculate the credibility on each set separately and generate the final credibility based on a discounting factor vector \(\beta\) using the following formula.

\[
fc = \beta_n c_n + \beta_{n-1} c_{n-1} + \ldots + \beta_1 c_1
\]

where \(n\) is the number of sets. The larger \(n\) is, the closer to the current time the set of records is. The discounting factor vector has the following properties:

- it is strictly monotonic, i.e. \(\beta_n > \beta_{n-1} > \ldots > \beta_1\);
- the sum of factors is 1, i.e. \(\sum \beta_i = 1\).

Monotonicity guarantees that the most recent credibility has the largest weight to determine the final credibility. The fact that the sum is set to 1 guarantees that there is no credibility loss due to discounting factors in the integration. Two examples of discounting factors are as follows:

- \(1/2 + 1/2^n, 1/4, \ldots, 1/2^n\);
- \(0.9 + 1 	imes 10^{-n}, 0.09, \ldots, 9 	imes 10^{-n}\).

VII. RELATED WORK

Proposals on annotation management [16], [17] are closely related to studies on provenance management [18], [19], [20], [21]. Most of these proposals focus on how to efficiently store and/or query annotations. Among these proposals on annotation management, [7] is probably the closest to our proposal because we share a same motivation application. However, [7] attacks a different problem by trying to answer questions like “does B believe A is correct” or “who has a conflicting belief on fact X from A”. It is unsurprising that authors of [7] adopt a modal logic-based approach to address these questions. However, there is no solution given in [7] to answer questions like “Is A more correct than B?” that is addressed by our approach.

Work on reputation management [22], [15] in P2P network evaluated the reputation value of peers based on scores from other peers. Work on Wikipedia trust [23], [24], [25] evaluated the trustworthiness of authors and content based on different factors, e.g. number of edits. However, none of these approaches considered subjective confidence degrees and ontological credibility.

Several approaches have been proposed to manage uncertain and incomplete information in terms of probabilistic databases [26], [27], [28], [29], [30], [31]. However, we are not focusing on the evaluation, query, and storage of uncertain information; we apply uncertain confidence degrees to weight the credibility of contributors.

There are also approaches [32], [33], [34] to rank the integrated biological data. These proposals rank data mainly based on the link structure information during the integration process. Our approach, by contrast, ranks correctness mainly based on the confidence degrees and the credibility of contributors.

VIII. CONCLUSION

In this paper, we propose the concept of credibility-enhanced curated database and an algorithm to evaluate the correctness of annotations and source information in curated databases. We also carry out a comprehensive simulation to verify the efficiency and effectiveness of the algorithm and investigate the quantification of subjective confidence degrees. Our approach can be applied to other annotation scenarios such as the online review score systems and improve the quality of these scores. In the future, we plan to investigate additional measures that may further improve our propagation algorithm, e.g., the measure for partial correctness. Partial correctness is useful for imprecise concepts in sources or annotations that contain a correct concept. We may further investigate the credibility building problem in more complicated scenarios, e.g., there are malicious contributors in resellerrating.com.