A WOWA-based Aggregation Technique on Trust Values Connected to Metadata[1]

Ernesto Damiani[2] Sabrina De Capitani di Vimercati[3]
Pierangela Samarati[4] Marco Viviani[5]

Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano
Via Bramante, 65 – 26013 Crema (CR) – Italia

Abstract

Metadata produced by members of a diverse community of peers tend to contain low-quality or even mutually inconsistent assertions. Trust values computed on the basis of users' feedback can improve metadata quality and reduce inconsistency, eliminating untrustworthy assertions.

In this paper, we describe an approach to metadata creation and improvement, where community members express their opinions on the trustworthiness of each assertion. Our technique aggregates individual trustworthiness values to obtain a community-wide assessment of each assertion. We then apply a global trustworthiness threshold to eliminate some assertions to reduce the metadatabase's overall inconsistency.

Key words: Metadata assertions, trustworthiness, community-wide trust.

1 Introduction

The growing need of share and manage knowledge about data is strictly connected to the interest in studying and developing systems for generating and managing metadata. Typically, metadata provide annotations specifying content, quality, type, creation, and spatial information of a data item. Though

[1] This work was supported in part by the European Union within the PRIME Project in the FP6/IST Programme under contract IST-2002-507591 and by the Italian MIUR within the KIWI and MAPS projects.

[2] Email: damiani@dti.unimi.it

[3] Email: decapita@dti.unimi.it

[4] Email: samarati@dti.unimi.it

[5] Email: viviani@dti.unimi.it

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

a number of specialized formats based on Resource Description Framework (RDF) are available, metadata can be stored in any format such as a text file, eXtensible Markup Language (XML), or database record. There are a number of advantages in using information extracted from data instead of data themselves. First of all, because of their small size compared to the data they describe, metadata are more easily shareable than data. Thanks to metadata shareability, information about data becomes readily available to anyone seeking it. Thus, metadata make data discovery easier and reduces data duplication.

On the other hand, metadata can be generated by a number of sources (the data owner, other users, automatic tools) and may or may not be digitally signed by their author. Therefore, metadata have non-uniform *trustworthiness*. To take full advantage of metadata, it is fundamental that (*i*) users are aware of each metadata level of trustworthiness, (*ii*) metadata trustworthiness is continuously updated, for example, based on the view of the user community. This is even more important when the original sources of metadata are automatic metadata generators whose error rates are not negligible, or when metadata are created via peer certifications, that is, via assertions by users that other users belong to a category or have a given property. In some systems (e.g., Advogato [12]), peer certifications are input to a trust metric, whose output is a simple Boolean value for each user. Here, we are interested in a more general case, when metadata consists of generic assertions that may or may not be peer certifications. Also, in this paper we do not impose any constraint on the schema for these assertions. Rather, we plan to use a fuzzy metric of trustworthiness to improve the overall quality of a metadata base composed of independently generated assertions. Metadata produced by members of a diverse community of peers tend to contain sets of mutually inconsistent assertions. A simple example is the triple "X belongs to class A", "X belongs to class B" and "class A is the complement of class B". In our approach, inconsistency is not detected or corrected via reasoning.[6] Rather, we leave it to community members to express their views on the trustworthiness of each assertion in the metadata base. Then, we aggregate individual trustworthiness values to obtain a community-wide assessment of each assertion. Finally, we apply a trustworthiness threshold to eliminate some assertions, hopefully reducing the overall inconsistency of the metadatabase. In this paper, our aim is to describe an aggregation function to compute community-wide trust values on metadata. Our aggregator is able to take into consideration various aspects connected to human behavior when accessing metadata.

The remainder of this paper is organized as follows. Section 2 discusses how different trust values can be aggregated. Section 3 presents the architecture

[6] Detecting inconsistencies by comparing assertions poses several practical problems due to different ways of expressing the same assertion. For instance, should "X" be an ASCII string, containing the name of the resource, a cryptographic hash of the resource itself or the URL where the resource can be downloaded?

of our proof-of-concept prototype together with the aggregation technique. Finally, Section 4 concludes the paper.

2 A fuzzy-based approach for trust aggregation

There are two main problems in managing trustworthiness values about metadata assertions: how to collect them and how to aggregate them. Several different approaches to collecting ratings have been proposed, based on *implicit* and/or *explicit behavior* of users. Traditionally, research approaches [4,9] distinguish between two main types of trust management systems, namely *Centralized Reputation Systems* and *Distributed Reputation Systems.*[7]

In centralized reputation systems, trust information is collected from members in the community in the form of *ratings* on resources. The central authority collects all the ratings and derives a score for each resource.

In a distributed reputation system there is no a central location for submitting ratings and obtaining resources' reputation scores; instead, there are distributed stores where ratings can be submitted. In a "pure" P2P setting, each peer has its own repository of trust values. In both cases, initial metadata trust values (often produced through a non completely trusted production process) can be modified based on users' navigation activity. Information on user behavior can be captured and transformed in a metadata layer expressing the trust degree related to the single assertion.[8]

Ratings based on implicit user behavior can take into account the time spent by each user working on the resource. On the other hand, users can express their views on metadata describing a resource by casting a vote on each assertion's trustworthiness. These explicit votes are usually cast by a small subset of the users, depending on their role or their expertise.

The main problem to solve, once trust values have been collected, is their aggregation. First of all, it is necessary to take into account the level of anonymity provided by the system. If all users are anonymous, all votes contribute in the same way to the overall trust value on metadata assertions about a resource. When users have an identity, ratings have to be aggregated initially at user level, and then at global level. Several techniques for computing reputation and trust measures in non-anonymous environments have been proposed [10] that can be briefly summarized as follows.

- *Summation or Average of ratings.* This technique computes the sum of positive and negative ratings separately, obtaining the total score by subtracting the negative votes from the positive ones.

[7] There is a clear distinction between *trust* and *reputation*. In general, a trust value T can be computed based on its reputation R, that is, $T = \phi(R, t)$, where t is the time elapsed since when the reputation was last modified [3].

[8] While we shall not deal with metadata signing in this paper, it is important to remember that assertions can be digitally signed.

- *Bayesian Systems.* This kind of systems take binary ratings as input and compute reputation scores by statistically updating beta probability density functions (PDF). They are particularly used in *recommender systems* [2].
- *Discrete Trust Models.* These models rely on human verbal statements to rate assertions (e.g., *Very Trustworthy*, *Trustworthy*, *Untrustworthy*, *Very Untrustworthy*).
- *Belief Models.* Belief theory is a framework related to probability theory, but where the sum of beliefs over all possible outcomes does not necessarily add up to 1. The missing part is interpreted as uncertainty.
- *Fuzzy Models.* Linguistic variables can be used to represent trust; in this case, membership functions describe to what degree an assertion can be described as trustworthy or not trustworthy.
- *Flow Models.* They compute trust by transitive iteration through looped or arbitrarily long chains.

Our approach computes the level of trust of an assertion as the aggregation of multiple *fuzzy values* representing trust resulting from human interactions with metadata assertions. In principle, this method could be used to assign a trust value to any kind of data [6], but in this case data semantics may not be clear enough to support automatic choice of the correct aggregation operator to be used. Knowledge about the context connected to data is fundamental so the choice of the correct operator is strongly context-dependent.

Dealing with metadata, the choice of the aggregation operator for trust values remains crucial, but metadata richer semantics makes this work easier. A simple arithmetic average would perform a rough compensation between high and low values; for this reason we turned to the *Weighted Mean* [1] and the *Ordered Weighted Averaging operator* [14], whose behavior has been analyzed in [7,8]. The difference between these two functions is in the meaning they assign to weights that have to be combined with input values. Weighted means aggregate trust values from different sources, taking into account the reliability of each source. On the other hand, the OWA operator weights trust values in relation to their size, without taking into account which sources have expressed them.

Our solution represents an alternative to statistical or probabilistic approaches (e.g., Bayesian Systems or Belief Methods). In [3] an OWA-based solution is compared with a probabilistic approach, called *EigenTrust* [11], where a reputation is defined as a probability and can be computed by an event-driven method using a Bayesian interpretation. From this comparison results that a major advantage of fuzzy aggregation techniques is their speed. In [3] we showed that, even if the fuzzy solution has a slower start-up, the global convergence speed is faster than the EigenTrust algorithm.

In non-anonymous scenarios, different users or roles express votes on an assertion. Therefore, it is important to weight each vote according to the reliability of the user who casts it (e.g., based on her profile). Furthermore,

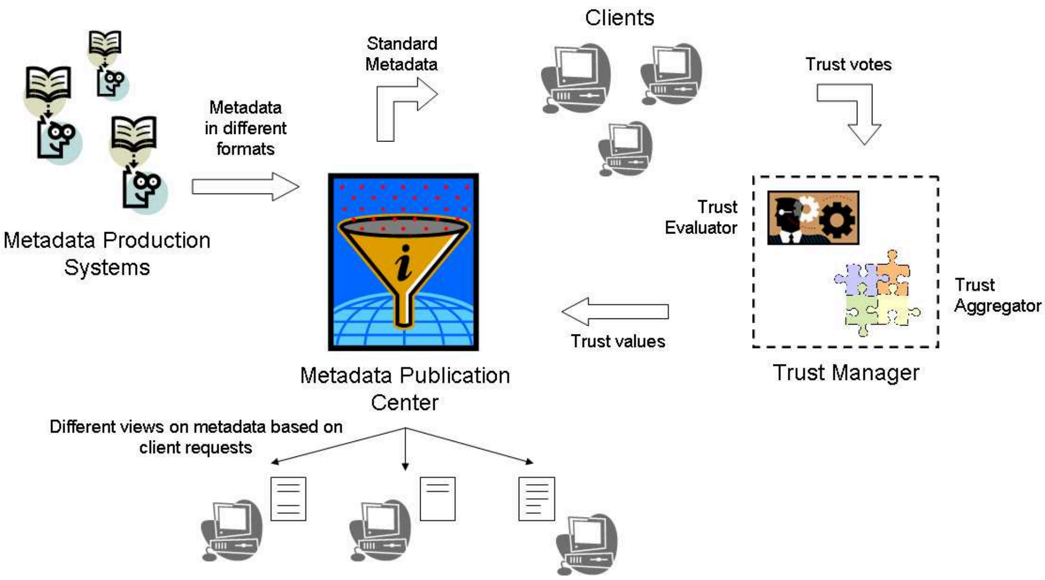

Clients
Standard Metadata
Trust votes
Metadata in different formats
Metadata Production Systems
Trust Evaluator
Trust Aggregator
Trust values
Trust Manager
Metadata Publication Center
Different views on metadata based on client requests

Fig. 1. The architecture of our prototype

votes can be aggregated depending on a number of other criteria (e.g., user location, connection medium, and so on).

In the following Section, we briefly describe the architecture of our proof-of-concept prototype and present the aggregation function that takes into consideration the above-mentioned aspects.

3 Architecture and aggregation function

A proof-of-concept prototype has been designed and implemented to validate the proposed method. Figure 1 illustrates the architecture of our prototype. A centralized *Metadata Publication Center* (MPC) collects and displays metadata assertions, possibly in different formats and coming from different sources. The MPC can be regarded as a semantic search engine, containing metadata provided by automatic generators crawling the web or by other interested parties. The MPC will assign different trust values to assertions depending on their origin: assertions manually provided by a domain expert are much more reliable than assertions automatically generated and submitted by a crawler.

The metadata in the MPC are indexed and a group of *Clients* interacts with the metadata by navigating them and providing implicitly (with their behavior) or explicitly (by means of an explicit vote) an evaluation about metadata trustworthiness. This trust-related information is passed by the MPC to the *Trust Manager*. The TM is composed of two modules:

- the *Trust Evaluator* examines metadata and evaluates their reliability;
- the *Trust Aggregator* aggregates all inputs coming from the Trust Evaluator clients according to a suitable aggregation function (in our prototype, the

WOWA operator, as discussed in the next subsection).

The Trust Manager is the computing engine behind the MPC module that provides interested parties with a visual overview on the metadata reliability distribution.

We are now ready to present the function used for aggregating trust values.

3.1 The WOWA operator

The *Weighted OWA operator* (WOWA) [13] is a promising approach, because it combines the advantages of both the OWA operator and the weighted mean. WOWA uses two sets of weights: **p** corresponds to the *relevance of the sources*, and **w** corresponds to the *relevance of the values*.

Definition 3.1 Let **p** and **w** be two weight vectors of dimension n ($\mathbf{p} = [p_1\ p_2\ ...\ p_n]$, $\mathbf{w} = [w_1\ w_2\ ...\ w_n]$) such that: *i)* $p_i \in [0,1]$ and $\sum_i p_i = 1$; *ii)* $w_i \in [0,1]$ and $\sum_i w_i = 1$. A mapping $f_{\text{WOWA}} : \mathbb{R}^n \rightarrow \mathbb{R}$ is a *Weighted Ordered Weighted Averaging* (WOWA) operator of dimension n if

$$f_{\text{WOWA}}(a_1, a_2, ..., a_n) = \sum_i \omega_i a_{\sigma(i)} \tag{1}$$

where $\{\sigma(1), \sigma(2), ..., \sigma(n)\}$ is a permutation of $\{1, 2, ..., n\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}, i = 2, ..., n$, weight ω_i is defined as

$$\omega_i = w^* \left(\sum_{j \leq i} p_{\sigma(j)} \right) - w^* \left(\sum_{j < i} p_{\sigma(j)} \right) \tag{2}$$

with w^* a monotonic function (e.g., a polynomial) that interpolates the points $(i/n, \sum_{j \leq i} w_j)$ together with the point $(0,0)$. Term ω denotes the set of weights $\{\omega_i\}$, that is, $\omega = \{\omega_1, \omega_2, ..., \omega_n\}$.

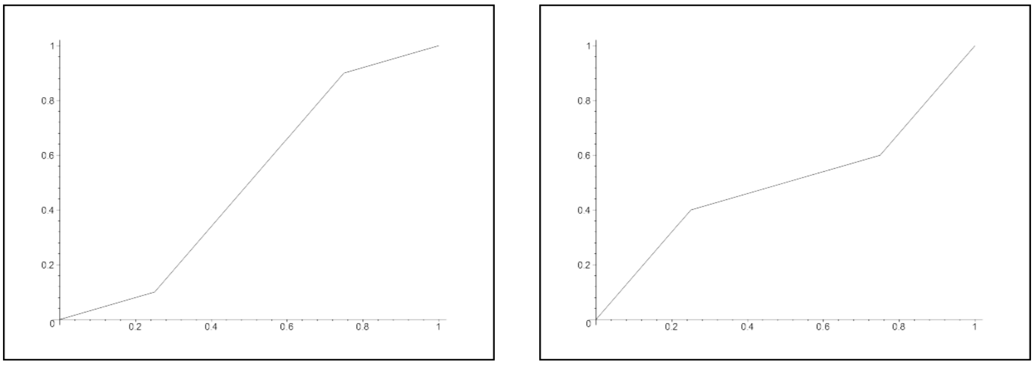
Compared with plain OWA, the WOWA operator allows to model more complex situations, for example, a scenario in which not all voters are equal, regardless of the vote they express.

Selecting the WOWA operator

Given two weight vectors **p** and **w** and a data vector **a**, let $S = \{(i/n, \sum_{j \leq i} w_j) \mid i = 1, ..., n\} \cup \{0, 0\}$. According to Definition 3.1, we need to define the function w^* interpolating S. To this purpose, two possible approached can be applied:

- we first define vector **w** and then function w^* is established;
- we first define function w^*.

In the first approach, the monotonic function w^* can be obtained by applying any method that, starting from the monotonic data points in the unit interval, defines a monotonic, bounded function.



(a) (b)

Fig. 2. The interpolation functions w'^* (a) and w''^* (b)

In the second approach, the set of weights ω is derived from w^*, where w^* is any monotonically increasing function within the $[0, 1]$ interval with $w^*(0) = 0$ and $w^*(1) = 1$.

Referring to the first method, suppose that we have a set of input values $\mathbf{a} = [a_1\ a_2\ a_3\ a_4]$ already listed in a decreasing order. The weighting vector $\mathbf{w}$ is determined according to the importance that we assign to the data values. For instance, vector $\mathbf{w}' = [.1\ .4\ .4\ .1]$ indicates that central values are more important than extreme ones; by contrast, vector $\mathbf{w}'' = [.4\ .1\ .1\ .4]$ indicates that extreme values are the most important. Figure 2(a) and Figure 2(b) illustrates the corresponding interpolation functions w'^* and w''^*, respectively[9] .

A correct definition of the vector $\mathbf{p}$ allows us to take into consideration trust values reliability depending on their information source. For instance, by taking vector $\mathbf{w}' = [.1\ .4\ .4\ .1]$ (or function w'^*) and $\mathbf{p}' = [.15\ .35\ .35\ .15]$ (where central values are considered to be more reliable), it is easy to see that the weights ω_1 and ω_2 (Equation 2) grow with w_i, because the weighting vector $\mathbf{w}$ enhances central values. As another example, vector $\mathbf{p}'' = [.35\ .15\ .15\ .35]$ leads to decreasing central ω_i because central values are considered less reliable then extreme ones.

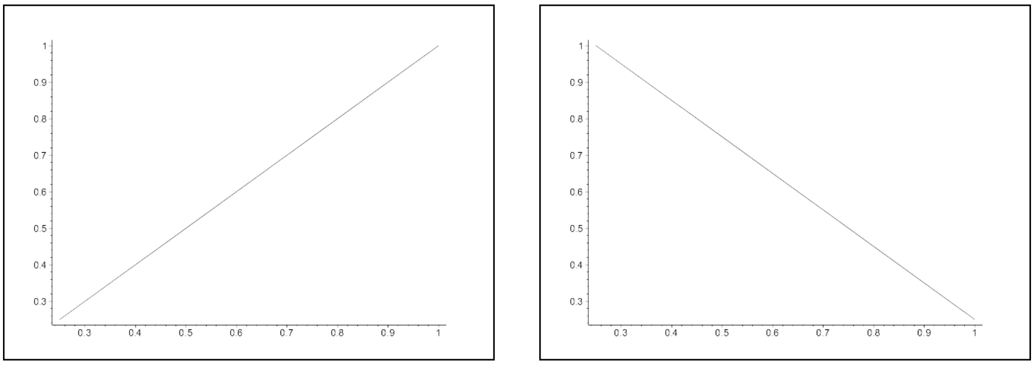
3.2 An Example

We now give a worked-out example illustrating two different approaches in assigning relevance to the data values. For the sake of simplicity, we assume that array $\mathbf{w}$ is composed of values of the form $w_i = \frac{k}{n}$. The following two cases are then considered:

Diffident. Variable k ranges from 1 to n (Figure 3(a)).

Confident. Variable k ranges from n to 1 (Figure 3(b)).

[9] Note that, following the second approach, it is possible to first define the interpolation function and then to extract the weighting vector.



(a) (b)

Fig. 3. Weights w_i in the diffident approach (a) and in the confident approach (b)

The diffident approach.

Looking at Figure 3(a) and considering the fact that the WOWA operator is applied to a permutation of data values a_i $\big($where $a_{\sigma(i-1)} \geq a_{\sigma(i)}, \forall i = 2, ..., n\big)$ it is easy to understand why the first approach is called "diffident": it reduces the impact of high trust values. The weights $w_i = k/n, k = 1, \ldots, n$, have to be normalized by dividing them by their sum:

$$w_i = \frac{k/n}{\sum_{k=1}^{n} k/n} \; .$$

Suppose now that $\mathbf{a} = [.9\ .7\ .5\ .3\ .1]$ (the values are already ordered), $\mathbf{p} = [\frac{1}{15}\ \frac{4}{15}\ \frac{2}{15}\ \frac{5}{15}\ \frac{3}{15}]$, and $\mathbf{w} = [\frac{1}{5}\ \frac{2}{5}\ \frac{3}{5}\ \frac{4}{5}\ 1]$. We obtain the normalized vector $\mathbf{w_n} = [\frac{1}{5}\ \frac{2}{5}\ \frac{3}{5}\ \frac{4}{5}\ 1]/\frac{15}{5} = [\frac{1}{15}\ \frac{2}{15}\ \frac{3}{15}\ \frac{4}{15}\ \frac{5}{15}]$. The next step consists in finding a function w^* interpolating the points $(i/n, \sum_{j \leq i} w_j)$:

$i = 1 \quad \big(\frac{1}{5}, w_1\big) = \big(\frac{1}{5}, \frac{1}{15}\big)$

$i = 2 \quad \big(\frac{2}{5}, w_1 + w_2\big) = \big(\frac{2}{5}, \frac{3}{15}\big) = \big(\frac{2}{5}, \frac{1}{5}\big)$

$i = 3 \quad \big(\frac{3}{5}, w_1 + w_2 + w_3\big) = \big(\frac{3}{5}, \frac{6}{15}\big) = \big(\frac{3}{5}, \frac{2}{5}\big)$

$i = 4 \quad \big(\frac{4}{5}, w_1 + w_2 + w_3 + w_4\big) = \big(\frac{4}{5}, \frac{10}{15}\big) = \big(\frac{4}{5}, \frac{2}{3}\big)$

$i = 5 \quad (1, w_1 + w_2 + w_3 + w_4 + w_5) = (1, 1)$

The equation of the interpolation function (Figure 4) is:

$$w^*(x) = \frac{5}{6}x^2 + \frac{1}{6}x \; .$$

We use $w^*(x)$ to compute the final weights ω_i:

$i = 1 \quad \omega_1 = w^*(p_1) = w^*\left(\frac{1}{15}\right) = \frac{2}{135}$

$i = 2 \quad \omega_2 = w^*\left(\sum_{i=1}^{2} p_i\right) - w^*(p_1) = w^*\left(\frac{1}{3}\right) - w^*\left(\frac{1}{15}\right) = \frac{2}{15}$

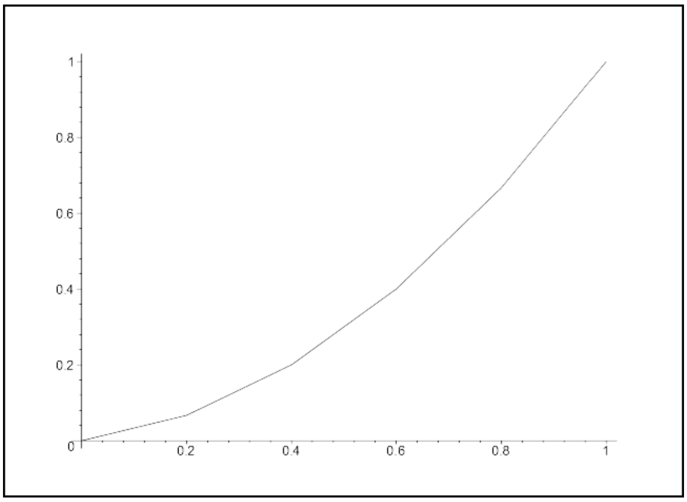


Fig. 4. The interpolation function w^* in the *diffident approach*

$i = 3 \quad \omega_3 = w^*\left(\sum_{i=1}^{3} p_i\right) - w^*\left(\sum_{i=1}^{2} p_i\right) = w^*\left(\frac{7}{15}\right) - w^*\left(\frac{1}{3}\right) = \frac{1}{9}$

$i = 4 \quad \omega_4 = w^*\left(\sum_{i=1}^{4} p_i\right) - w^*\left(\sum_{i=1}^{3} p_i\right) = w^*\left(\frac{4}{5}\right) - w^*\left(\frac{7}{15}\right) = \frac{11}{27}$

$i = 5 \quad \omega_5 = w^*\left(\sum_{i=1}^{5} p_i\right) - w^*\left(\sum_{i=1}^{4} p_i\right) = w^*(1) - w^*\left(\frac{4}{5}\right) = \frac{1}{3}$

We are now ready to compute the final value of f_{WOWA} (σ is the identical permutation):

$$f_{\mathrm{WOWA}}(.9, .7, .5, .3, .1) = \sum_{i=1}^{5} \omega_i a_i = .31\overline{7}\ .$$

Note that, in this example, the simple OWA method introduced in [3] corresponds to equation $\sum_{i=1}^{5} w_i a_i$. It is immediate to see that the result is only influenced by the ordering of the vector $\mathbf{a}$ and by the relevance values of the weighting vector $\mathbf{w}$ and that the relevance of the sources providing the trust values is not considered at all. In this case, the final result would be $f_{\mathrm{OWA}}(.9, .7, .5, .3, .1) = \sum_{i=1}^{5} w_i a_i = .3\overline{6}$.

The confident approach.

We enhance the impact of high trust values in array $\mathbf{a}$ by choosing a function interpolating the points $(i/n, \sum_{j \leq i} w_j)$, where $w_i = \frac{k}{n}, k = n, n-1, \ldots, 1$. Operating in the same way as before on the weighted normalized vector $\mathbf{w_n} = [\frac{5}{15}\ \frac{4}{15}\ \frac{3}{15}\ \frac{2}{15}\ \frac{1}{15}]$, we obtain the w^* interpolating function:

$i = 1 \quad \left(\frac{1}{5}, w_1\right) = \left(\frac{1}{5}, \frac{5}{15}\right) = \left(\frac{1}{5}, \frac{1}{3}\right)$

$i = 2 \quad \left(\frac{2}{5}, w_1 + w_2\right) = \left(\frac{2}{5}, \frac{9}{15}\right) = \left(\frac{2}{5}, \frac{3}{5}\right)$

$i = 3 \quad \left(\frac{3}{5}, w_1 + w_2 + w_3\right) = \left(\frac{3}{5}, \frac{12}{15}\right) = \left(\frac{3}{5}, \frac{4}{5}\right)$

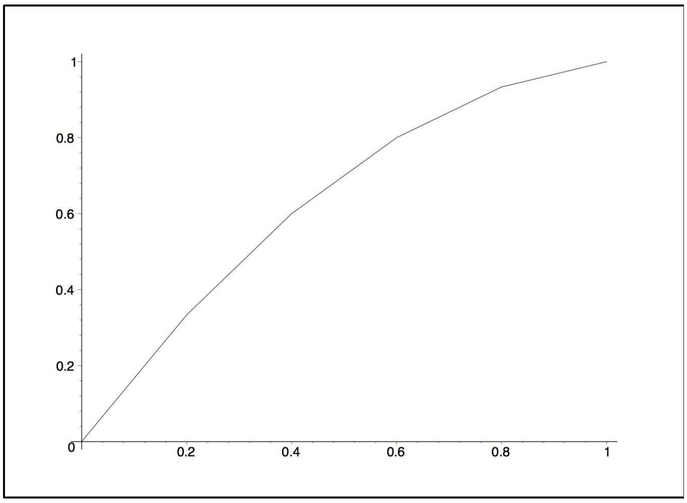


Fig. 5. The interpolation function w^* in the *confident approach*

$i = 4 \quad \left(\frac{4}{5}, w_1 + w_2 + w_3 + w_4\right) = \left(\frac{4}{5}, \frac{14}{15}\right)$

$i = 5 \quad (1, w_1 + w_2 + w_3 + w_4 + w_5) = (1, 1)$
The w^* interpolating function is expressed by the following equation (Figure 5):

$$w^*(x) = -\frac{5}{6}x^2 + \frac{11}{6}x \ .$$

The final weights ω_i are calculated as follows ($\mathbf{p}$ is the same vector used in the previous example):
$i = 1 \quad \omega_1 = w^*(p_1) = w^*\left(\frac{1}{15}\right) = \frac{16}{135}$

$i = 2 \quad \omega_2 = w^*\left(\sum_{i=1}^{2} p_i\right) - w^*(p_1) = w^*\left(\frac{1}{3}\right) - w^*\left(\frac{1}{15}\right) = \frac{2}{5}$

$i = 3 \quad \omega_3 = w^*\left(\sum_{i=1}^{3} p_i\right) - w^*\left(\sum_{i=1}^{2} p_i\right) = w^*\left(\frac{7}{15}\right) - w^*\left(\frac{1}{3}\right) = \frac{7}{45}$

$i = 4 \quad \omega_4 = w^*\left(\sum_{i=1}^{4} p_i\right) - w^*\left(\sum_{i=1}^{3} p_i\right) = w^*\left(\frac{4}{5}\right) - w^*\left(\frac{7}{15}\right) = \frac{7}{27}$

$i = 5 \quad \omega_5 = w^*\left(\sum_{i=1}^{5} p_i\right) - w^*\left(\sum_{i=1}^{4} p_i\right) = w^*(1) - w^*\left(\frac{4}{5}\right) = \frac{1}{15}$

As expected, the final trust value given by the computation of the function f_{WOWA} [10] is higher respect to the "diffident" approach:

$$f_{\text{WOWA}}(.9, .7, .5, .3, .1) = \sum_{i=1}^{5} \omega_i a_i \ = .54\overline{8} \ .$$

[10] As illustrated for the previous approach, the f_{OWA} would be $f_{\text{OWA}}(.9, .7, .5, .3, .1) = \sum_{i=1}^{5} w_i a_i = .6\overline{3}$.

4 Conclusions

We have presented a WOWA-based aggregation technique on individual trust values on metadata assertions. These assertions can be automatically generated by collecting user feedback in a non-intrusive way. Our technique allows to take several effects into account, including users' attitude and sources reliability. When assertions are associated with identities of annotators, our schema can readily be extended to deal with trust on annotators as well as on assertions [3]. It is also possible to envision many other usage scenarios, such as the one involving non-functional descriptors of web services published by service providers in a service-oriented software architecture. In this case, our approach could deal with the non-uniform reliability of service self-descriptions, which is liable to imprecision and may lead to unexpected results [5].

References

[1] Aczél, J.: *On Weighted synthesis of judgments*, Aequationes Math. **27** (1984) 288–307.

[2] Adomavicius, G., Tuzhilin, A.: *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*, IEEE Transactions on Knowledge and Data Engineering **17(6)** (2005) 734–749.

[3] Aringhieri, R., Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P.: *Fuzzy Techniques for Trust and Reputation Management in Anonymous Peer-to-Peer Systems*, Journal of the American Society for Information Science and Technology (JASIST) (to appear).

[4] Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.: *The role of trust management in distributed systems security*, Secure Internet Programming (1999) 79–97.

[5] Bosc, P., Damiani, E., Fugini, M.: *Fuzzy service selection in a distributed object-oriented environment*, IEEE Transactions on Fuzzy Systems Publication **9(5)** (2001) 682–698.

[6] Ceselli, A., Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., Samarati, P.: *Modeling and assessing inference exposure in encrypted databases*, ACM Transactions on Information and System Security **8(1)** (2005) 119–152.

[7] Fodor, J., Marichal, J. L., Roubens, M.: *Characterization of the Ordered Weighted Averaging Operators*, IEEE Transactions on Fuzzy Systems **3(2)** (1995) 236–240.

[8] Grabisch, M.: *Fuzzy integral in multicriteria decision making*, Fuzzy Sets and Systems **69** (1995) 279–298.

[9] Jøsang, A.: *The right type of trust for distributed systems*, Proceedings of the 1996 Workshop on New Security Paradigms (1996).

[10] Jøsang, A., Roslan, I., Boyd, C.: *A Survey of Trust and Reputation Systems for Online Service Provision*, Decision Support Systems (to appear) (2005).

[11] Kamvar, S., Schlosser, M., Garcia-Molina, H.: *The EigenTrust Algorithm for Reputation Management in P2P Networks*, Proceedings of the Twelfth International World Wide Web Conference (2003).

[12] Levien, R.: from Advogato Website, Advogato Trust Metric,
URL: http://www.advogato.org/trust-metric.html.

[13] Torra, V.: *The weighted owa operator*, International Journal of Intelligent Systems **12(2)** (1997) 153–166.

[14] Yager, R.: *On ordered weighted averaging aggregation operators in multicriteria decision making*, IEEE Transactions on Systems, Man and Cybernetics **18(1)** (1988) 183–190.

[15] Yager, R.: *Quantifier Guided Aggregation Using OWA operators*, International Journal of Intelligent Systems **11** (1996) 49–73.