A General Model for Similarity Measurement between Objects

Manh Hung Nguyen

[1]Posts and Telecommunications Institute of Technology (PTIT)
Hanoi, Vietnam

[2]UMI UMMISCO 209 (IRD/UPMC), Hanoi, Vietnam

Thi Hoi Nguyen

Vietnam Commercial University, Hanoi, Vietnam

Abstract—**The problem to detect the similarity or the difference between objects are faced regularly in several domains of applications such as e-commerce, social network, expert system, data mining, decision support system, etc. This paper introduces a general model for measuring the similarity between objects based on their attributes. In this model, the similarity on each attribute is defined with different natures and kinds of attributes. This makes our model is general and enables to apply the model in several domains of application. We also present the applying of the model into two applications in social network and e-commerce situations.**

Keywords—*object similarity; multiple attributes similarity; similarity measurement; decision support.*

I. Introduction

The problem to detect the similarity or the difference between objects are faced regularly in several domains of applications such as e-commerce, social network, expert system, data mining, decision support system, etc. There are many model proposed to measure the similarity between objects in these kinds of applications. For instances, D. Lin [3] proposed a model to measure the similarity between any two objects based on information-theoretic approach. Sayal and Kumar [10] proposed a model on clustering categorical attributes of relational data set types making use of the property of functional dependency as parameter to measure similarity. Reddy and Krishnaiah [9] proposed a similarity measure known as multi-viewpoint based similarity measure to ensure the clusters show all relationships among objects. Honko [1] proposed and investigated several similarity measures on complex structured objects. The objects are understood as examples of a target relation, and they are expressed in a first-order logic language. Meanwhile some other proposed some metrics to measure the similarity between profiles [8], [7]; similarity between objects based on images [5]; similarity between two trajectories [4]; or similarity between texts [6], [2], etc.

This paper introduces a general model for measuring the similarity between objects based on their attributes. In this model, the similarity on each attribute is defined with different kinds of attributes. This makes our model is general and enables to apply the model in several domains of application.

The paper is organised as follows: Section II presents the general similarity model. Section III presents some case studies for the proposed similarity model. Section IV is the conclusion and perspectives.

II. A Model for Measurement of Similarity between Objects

Without loss of generality, we assume that there are n concerned features $\{a^1, a^2, ...a^n\}$, which are attributes of considered object p, to measure the similarity between two objects. There are two steps as follows:

- Step 1: estimate the similarity on each considered feature and normalised it into the unit interval $[0, 1]$.

- Step 2: the similarity between two agents is then estimated by averaging the similarity between them on all considered features.

A. Similarity on each feature

The similarity on each considered feature of object is differently estimated based on the kind of feature. We distinguish five kind of feature:

- Feature whose value is a single number

- Feature whose value is a single string

- Feature whose value is an interval of number

- Feature whose value is a single matching

- Feature whose value is a set of ordered discrete numbers (a vector)

- Feature whose value is a set of non-ordered discrete numbers

- Feature whose value is a set of strings

Note that in case that the feature value is a kind of object, we could recursively apply this model (with two steps) to estimate the similarity between the two object to have the similarity of the feature.

1) Feature whose value is a single number: For this kind of feature, we define a possible interval, call $[MIN, MAX]$, for the value of the feature. It means that the value of the feature is acceptable if only if it is inside a given interval. Therefore, suppose that a_i^k, a_j^k are two single number values on the features a^k, of two objects i and j, respectively. The similarity between object i and j $(i, j \in A)$ on feature a^k is defined by the formula:

$$s_{ij}^k = 1 - \frac{\mid a_i^k - a_j^k \mid}{MAX - MIN} \tag{1}$$

For example, the feature of the *age* of a *seller* in an e-commerce application has a possible interval value of $[0, 100]$. So, if both seller i and j are 30 years old, then their similarity on the feature *age* is 1.00 (100%); if seller i is 30 years old, and seller j is 40 years old, then their similarity on the feature *age* is 0.90 (90%).

This computation is also applied for the feature whose value is a single date time.

2) Feature whose value is a single string: Suppose that a_i^k, a_j^k are two single string values on the features a^k, of two agents i and j, respectively. Let $length_i^k$, $length_j^k$ are the length of the single string value of the features a_i^k and a_j^k, respectively, counted by words. And $length_{ij}^k$ is the length of the longest sub-string between a_i^k and a_j^k, counted by word. The similarity between agent i and agent j ($i, j \in A$) on feature a^k is defined by the equation:

$$s_{ij}^k = \frac{2 * length_{ij}^k}{length_i^k + length_j^k}$$

For example, considering the feature *name* of two agents: "*Eton John*" (length = 2 words) and "*John Lennon*" (length = 2 words), the longest sub-string of these two names is "*John*" (length = 1), then their similarity on the feature *name* is $2 * 1 / (2 + 2) = 0.500$ (50.0%). Meanwhile, the similarity on the feature *name* of "*Eton John*" (length = 2) and "*John*" (length = 1) is $2 * 1 / (2 + 1) = 0.667$ (66.7%).

3) Feature whose value is a single matching: The value of this kind of feature could be a single number, single boolean value, or single string. But the matching is strictly binary: the similarity is 1 when the two values are identical; 0 when they are different.

Suppose that a_i^k, a_j^k are two single matching values on the features a^k, of two objects i and j, respectively. The similarity between object i and object j ($i, j \in A$) on feature a^k is defined by the formula:

$$s_{ij}^k = \begin{cases} 1 \text{ if } a_i^k = a_j^k \\ 0 \text{ if } a_i^k \neq a_j^k \end{cases} \qquad (2)$$

For example, considering the feature *original city* of user X is "*Paris*", the user Y is "*Paris*", and the user Z is "*London*", then similarity on the feature *original city* between X and Y is 1. Meanwhile, the similarity on the same feature between X and Z is 0.

4) Feature whose value is an interval of number: Suppose that $a_i^k = [x_1, x_2]$, $a_j^k = [y_1, y_2]$ are two interval values on the features a^k, of two objects i and j, respectively. And $[z_1, z_2]$ is the intersection interval of $[x_1, x_2]$ and $[y_1, y_2]$. The similarity between object i and object j ($i, j \in A$) on feature a^k is defined by the formula:

$$s_{ij}^k = \frac{2 * (z_2 - z_1)}{(x_2 - x_1) + (y_2 - y_1)} \qquad (3)$$

In the case that the intersection interval of $[x_1, x_2]$ and $[y_1, y_2]$ is empty, then $s_{ij}^k = 0$. This is also applied for the feature whose value is a time duration.

For example, considering the feature *price interval of preference* of a *seller* in an e-commerce application. If seller i

prefers the price between $\$100 - \$300$, and seller j prefers that between $\$200 - \$400$, then the intersection interval between them is $\$200 - \$300$, so the similarity on this feature between these two sellers is $2 * (300 - 200) / ((300 - 100) + (400 - 200)) = 0.50$ (50%).

5) Feature whose value is a set of ordered discrete numbers (a vector): Suppose that $a_i^k = (x_1, x_2, ...x_n)$, $a_j^k = (y_1, y_2, ...y_n)$ (n is the size of vector) are two vector values on the features a^k, of two objects i and j, respectively. And the value in each dimension of the vector is limited in an acceptable interval $[MIN, MAX]$. The similarity between object i and object j ($i, j \in A$) on feature a^k is defined by the formula:

$$s_{ij}^k = 1 - \frac{1}{n} \frac{\sum_{v=1}^{n} |x_v - y_v|}{(MAX - MIN)} \qquad (4)$$

For example, the feature *position* of a robot is represented in a 3-dimensions space whose the limit in each dimension is in an interval $[0, 10]$. If the robot i is at the position $(0, 3, 7)$, and robot j at the position $(6, 10, 2)$ then the similarity on the feature *position* between them is 0.40 (40%).

6) Feature whose value is a set of non-ordered discrete numbers: Suppose that $a_i^k = \{x_1, x_2, ...x_n\}$, $a_j^k = \{y_1, y_2, ...y_m\}$ (n,m are the size of set) are two set values on the features a^k, of two objects i and j, respectively. And the value in each element of the set is limited in an acceptable interval $[MIN, MAX]$. The similarity between object i and object j ($i, j \in A$) on feature a^k is estimated as follow:

- Sort the set a_i^k in increasing order such that $a_i^k = \{x_1' \leqslant x_2' \leqslant ... \leqslant x_n\}$

- Sort the set a_j^k in increasing order such that $a_j^k = \{y_1' \leqslant y_2' \leqslant ... \leqslant y_m\}$

- Without lost any generalisation, we suppose that $n \leqslant m$, the similarity between object i and object j ($i, j \in A$) on feature a^k is defined by the formula:

$$s_{ij}^k = 1 - \frac{1}{m} \left(\frac{\sum_{v=1}^{n} |x_v' - y_v'|}{(MAX - MIN)} + (m - n) \right) \qquad (5)$$

7) Feature whose value is a set of strings: Suppose that a_i^k, a_j^k are two values of type of set of strings on the features a^k, of two objects i and j, respectively. Let $size_i^k$, $size_j^k$ are the size of the set value of the features a_i^k and a_j^k, respectively. And $size_{ij}^k$ is the size of the intersection set of a_i^k and a_j^k. The similarity between object i and object j ($i, j \in A$) on feature a^k is defined by the formula:

$$s_{ij}^k = \frac{2 * size_{ij}^k}{size_i^k + size_j^k} \qquad (6)$$

For example, in the same application of e-commerce, the feature *favorite leisure* of seller i is a set of {*play football, travel, shopping*} (size = 3) and that of seller j is {*travel, play*

football, lecture, play tennis} (size = 4), then the intersection set of the two values is {*play football, travel*} (size = 2), so the similarity of i and j on this feature is $2*2/(3+4) = 0.57$ (57%).

It is easy to prove that all possible values of s_{ij}^k are lied in the interval $[0, 1]$. It means that, after this step, all similarities between the two agents on each feature are normalised into the unit interval. This normalisation enables us to avoid the domination of some feature whose value domain is vast on other features whose value domain is tight.

B. Similarity between objects

Once the similarities between two objects on each feature are estimated, the similarity between the two objects is then estimated by a weighted average aggregation the similarity between them on all considered features as follow:

- Sorting the similarities on each feature by the decreasing of the important level of the feature. Without lost of generalisation, the important level of considered features is decreased from feature 1 to feature n, then the similarities on each feature is $\{s_{ij}^1, s_{ij}^2, ...s_{ij}^n\}$.

- Choosing a weight vector $w = (w_1, w_2, ...w_n)$, where w_k is the weight of the kth sorted feature such that:

$$\begin{cases} w_{k_1} \geqslant w_{k_2} \text{ if } k_1 < k_2 \\ \sum_{k=1}^n w_k = 1 \end{cases} \qquad (7)$$

- The similarity between object i and object j is:

$$s_{ij} = \sum_{k=1}^n w^k * s_{ij}^k \qquad (8)$$

where w^k, s_{ij}^k are respectively the weight of the features a^k and the similarity on the feature a^k between object i and object j.

The usage of the weighted average operator leads this formula more flexible and generic. And the application designer could choose their own weight vector to customise the formula such that it is suitable for the nature of their application.

The weight vector is decreasing from head to tail. This corresponds to the decreased order of important level of sorted feature. This vector may be computed by means of Regular Decreasing Monotone (RDM) linguistic quantifier (Zadeh [12], Yager [11]) as follows:

The function $Q : [0, 1] \rightarrow [0, 1]$ is a Regular Decreasing Monotone linguistic quantifier, denote RDM, if and only if it satisfies the following conditions:

$$(i) \ Q(0) = 1$$
$$(ii) \ Q(1) = 0$$
$$(iii) \ Q(i_1) \geqslant Q(i_2) \text{ if } i_1 < i_2.$$

For example, the following functions are RDM:

$$(a) \ Q(x) = (1-x)^m \text{ with } m \geq 1$$
$$(b) \ Q(x) = 1 - \sqrt{1 - (1-x)^2}.$$

Suppose that Q is a RDM function, the vector w could be generated by function Q as follow:

$$w_i = Q\left(\frac{i-1}{n}\right) - Q\left(\frac{i}{n}\right) \text{ for } i = 1, ..., n$$

III. CASE STUDIES

In this section, we present the two potential applications of the proposed model: (i) applying the model to detect the similarity among user profiles in social network, and (ii) applying the model to choose the best product in an e-commerce system.

A. Detecting user profile similarity in social network

Nowadays, the rapid growth of social networks raises several related problems such as: how to find an account of an user on a social network that we know some pies of information about him; how to detect two (or more than two) accounts on two different social network are belong to an unique person; how to regroup the users of a social network into a set of separated groups;etc. This class of problems could be solved by applying the similarity model in this paper to estimate the similarity on user profiles of social networks.

For instance, let consider the problem to detect two (or more than two) accounts on two different social network are belong to an unique person. We could measure the similarity of each potential profile to the considered profile. The one having the highest similarity could be considered as the secondary profile of the considered user. In order to estimate the similarity of profiles, we consider a profile for *social network user* with following features, in the decreased order of important level:

- *Name*: The full display name of user. This feature is a kind of single string value.

- *Age*: The age of user. This feature is a kind of single number value.

- *Sex*: The sex of the user. This feature is a kind of single matching value.

- *Leisure of favorite*: The user preference of leisure. This feature is a kind of string set value.

- *Original city*: The original city of user. This feature is a kind of single matching value.

- *Work place*: The name of the company (or school) that the user is working for (or studying in, respectively). This feature is a kind of single matching value.

Assume that we are considering an user with profile attribute values are $p_0 = $ (*"Eton John"*, 62, male, {*music, cinema*}, *"London"*, *"Global Music"*). And we need to estimate the similarity of five profiles to that of Eton John as follow:

$p_1 = $ (*"John Eton"*, 60, male, {*music, sport*}, *"London"*, *"World Music"*)

$p_2 = $ (*"Eton John"*, 62, male, {*cinema, sport*}, *"London"*, *"Global Cinema"*)

$p_3 = $ (*"Eton John"*, 65, male, {*sport*}, *"London"*, *"Global Music"*)

TABLE I: Summary of similarity of five profiles compared to the considered profile p_0

Profiles	Name	Age	Sex	Leisure	City	Work place	Similarity
p_0	Eton John	62	male	music, cinema	London	Global Music	
p_1	John Eton	60	male	music, sport	London	World Music	0.74
	0.5	0.98	1	0.5	1	0	
p_2	Eton John	62	male	cinema, sport	London	Global Cinema	0.90
	1	1	1	0.5	1	0	
p_3	Eton John	65	male	sport	London	Global Music	0.85
	1	0.97	1	0	1	1	
p_4	Eton John	55	male	music, cinema	New York	Global Music	0.90
	1	0.93	1	1	0	1	
p_5	Eton John	60	male	cinema, sport	London	Global Music	0.93
	1	0.98	1	0.5	1	1	

$p_4 = $ (*"Eton John", 55, male, {music, cinema}, "New York", "Global Music"*)

$p_5 = $ (*"Eton John", 60, male, {cinema, sport}, "London", "Global Music"*)

The model is applied as follow (Table I):

- Choosing the RDM function $Q(x) = (1 - x)^2$ to generate the weight vector of six elements ($n = 6$) corresponding to six considered attributes of profile. Therefore, the values of $Q(0/6)$ to $Q(6/6)$ are: 1, 0.69, 0.44, 0.25, 0.11, 0.03, 0.

- The weight vector is thus: $w = (0.31, 0.25, 0.19, 0.14, 0.08, 0.03)$.

- Comparing the similarities: $s_5 > s_4 \sim s_2 > s_3 > s_1$, so the profile p_5 is considered as the most similar to the considered profile.

B. Choosing the best product in e-commerce

Let consider an e-commerce application of type e-market: there are several sellers in the e-market. Each of them sell several products. Each product has a different set of value on its attributes. There is a buyer who want to buy a product. He has some preference values on each attribute of product. The buyer could contact all sellers in the e-market to ask them to propose some products which satisfy his preference. Assume that each seller proposes at least one product to the buyer. So the buyer receives many potential products of his preference. But he have to choose only one product to buy. The question is which proposed product is the best suitable for the buyer, in regarding his preference?

For this problem, we could apply the proposed model as follow:

- Representing the preference product of the buyer on its preference values on n considered attributes as $p_0 = a_1^0, a_2^0, ... a_n^0$

- Assume that there are N products received from sellers. For each received product $p_i, i = 1..N$, representing it via its attribute values as $p_i = a_1^i, a_2^i, ... a_n^i$

- Estimating the similarity s_{i0} of each received product $p_i, i = 1..N$ with the preference product p_0. This step could be done by applying the proposed model in this paper.

- The received product $k, 1 \leq k \leq N$ which has the highest similarity s_{k0} with the preference product p_0

will be considered as the best suitable product for the given buyer.

For example, the exchanged product in the e-market is laptop, and the buyer consider on a set of eight attributes, in the decreased order of important level, in the personnel point of view of the buyer:

- *price*: The lower the price, the better for buyer. The highest acceptance threshold for this buyer is $500. The value of this attribute is single number, but the preference value is an interval of [0,500].

- *trademark*: The value of this attribute is single string. The the preference value is a set of strings (*Apple, Dell, Sony*).

- *processor speed*: The higher the processor speed, the better for buyer. The lowest acceptance threshold for this buyer is $3.0 GHz$. The value of this attribute is single number, but the preference value is an interval of [3.0,...].

- *RAM capacity*: The higher the RAM capacity, the better for buyer. The lowest acceptance threshold for this buyer is $2.4 GB$. The value of this attribute is single number, but the preference value is an interval of [2.4,...].

- *hard disk capacity (HDD)*: The higher the hard disk capacity, the better for buyer. The lowest acceptance threshold for this buyer is $100 GB$. The value of this attribute is single number, but the preference value is an interval of [100,...].

- *weight*: The lower the weight, the better for buyer. The highest acceptance threshold for this buyer is $3.5 kg$. The value of this attribute is single number, but the preference value is an interval of [0, 3.5].

- *screen size*: The higher the screen size, the better for buyer. The lowest acceptance threshold for this buyer is $15 inches$. The value of this attribute is single number, but the preference value is an interval of [15,...].

- *color*: The value of this attribute is single string. The the preference value is a set of strings (*Black, White*).

In summary, the preference values on attributes are $p_0 = $ (≤ 500, {*Apple, Dell, Sony*}, ≥ 3.0, ≥ 2.4, ≥ 100, ≤ 3.5, ≥ 15, {*Black, White*}). Assume that there are five received products as follow:

TABLE II: Summary of similarity of five products compared to the considered product p_0

Products	Price	Trademark	P. speed	RAM	HDD	Weight	S. size	Color	Similarity
p_0	≤ 500	Apple, Dell, Sony	≥ 3.0	≥ 2.4	≥ 100	≤ 3.5	≥ 15	Black, White	
p_1	400	Sony	2.8	2.2	100	2.5	14	Black	0.65
	1	1	0	0	1	1	0	1	
p_2	600	Dell	3.0	2.4	100	3.5	14	Black	0.73
	0	1	1	1	1	1	0	1	
p_3	700	Apple	3.5	2.8	150	2.0	15	White	0.77
	0	1	1	1	1	1	1	1	
p_4	500	Acer	3.0	2.4	100	3.8	15	Red	0.69
	1	0	1	1	1	0	1	0	
p_5	500	Apple	2.8	2.4	80	3.5	14	White	0.68
	1	1	0	1	0	1	0	1	

$p_1 = (400,\ Sony,\ 2.8,\ 2.2,\ 100,\ 2.5,\ 14,\ Black)$

$p_2 = (600,\ Dell,\ 3.0,\ 2.4,\ 100,\ 3.5,\ 14,\ Black)$

$p_3 = (700,\ Apple,\ 3.5,\ 2.8,\ 150,\ 2.0,\ 15,\ White)$

$p_4 = (500,\ Acer,\ 3.0,\ 2.4,\ 100,\ 3.8,\ 15,\ Red)$

$p_5 = (500,\ Apple,\ 2.8,\ 2.4,\ 80,\ 3.5,\ 14,\ White)$

The model is applied as follow (Table.II):

- Choosing the RDM function $Q(x) = (1-x)^2$ to generate the weight vector of eight elements ($n = 8$) corresponding to eight considered attributes of the product. Therefore, the values of $Q(0/8)$ to $Q(8/8)$ are: 1, 0.77, 0.56, 0.39, 0.25, 0.14, 0.06, 0.02, 0.

- The weight vector is thus: $w = (0.23, 0.21, 0.17, 0.14, 0.11, 0.08, 0.04, 0.02)$.

- Comparing the similarities: $s_3 > s_2 > s_4 > s_5 > s_1$, so the product p_3 is considered as the best suitable product to the buyer preference.

IV. CONCLUSIONS

In this paper, we present a model for estimating the semantic similarity between two objects based on their attributes or features via two steps. Firstly, the model estimates the similarity, between the two objects, on each feature and the results are normalised into the interval [0,1]. Secondly, the similarity between the two object is estimated by a weighted aggregation from the similarities on all considered features. This model could be applied into several applications to help some member of the system to choose the best suitable object from a set of potential objects considered such as find the most closed user profile in social network, choose the best product in an e-commerce application.

In the near future we will extend this model to compare the similarity between the behavior of users.

REFERENCES

[1] Piotr Honko. Description and classification of complex structured objects by applying similarity measures. *International Journal of Approximate Reasoning*, 49(3):539–554, 2008.

[2] Glen Jeh and Jennifer Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 538–543, New York, NY, USA, 2002. ACM.

[3] Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[4] Hechen Liu and Markus Schneider. Similarity measurement of moving object trajectories. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming*, IWGS '12, pages 19–22, New York, NY, USA, 2012. ACM.

[5] E. Nowak and F. Jurie. Learning Visual Similarity Measures for Comparing Never Seen Objects. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[6] Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, and Katia P. Sycara. Semantic matching of web services capabilities. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, ISWC '02, pages 333–347, London, UK, 2002. Springer-Verlag.

[7] Olga Peled, Michael Fire, Lior Rokach, and Yuval Elovici. Entity matching in online social networks. *Social Computing / IEEE International Conference on Privacy, Security, Risk and Trust, 2010 IEEE International Conference on*, 0:339–344, 2013.

[8] Elie Raad, Richard Chbeir, and Albert Dipanda. User profile matching in social networks. In *Proceedings of the 2010 13th International Conference on Network-Based Information Systems*, NBIS '10, pages 297–304, Washington, DC, USA, 2010. IEEE Computer Society.

[9] Gaddam Saidi Reddy and Dr.R.V.Krishnaiah. A novel similarity measure for clustering categorical data sets. *IOSR Journal of Computer Engineering (IOSRJCE)*, 4(6):37–42, 2012.

[10] Rishi Sayal and V. Vijay Kumar. A novel similarity measure for clustering categorical data sets. *International Journal of Computer Applications*, 17(1):25–30, March 2011. Published by Foundation of Computer Science.

[11] R. R. Yager. Nonmonotonic OWA operators. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 3(3):187–196, 1999.

[12] L. A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. pages 149–184, 1983.