A Similarity Measure on Tree Structured Business Data

Dianshuang Wu, Guangquan Zhang, Jie Lu
Decision Systems & e-Service Intelligence Lab, Centre for Quantum Computation & Intelligent Systems
Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia
Email: Dianshuang.Wu@student.uts.edu.au, Guangquan.Zhang@uts.edu.au, Jie.Lu@uts.edu.au

Wolfgang A. Halang
Chair of Computer Engineering, Fernuniversität, 58084 Hagen, Germany
Email: wolfgang.halang@fernuni-hagen.de

Abstract

In many business situations, products or user profile data are so complex that they need to be described by use of tree structures. Evaluating the similarity between tree-structured data is essential in many applications, such as recommender systems. To evaluate the similarity between two trees, concept corresponding nodes should be identified by constructing an edit distance mapping between them. Sometimes, the intension of one concept includes the intensions of several other concepts. In that situation, a one-to-many mapping should be constructed from the point of view of structures. This paper proposes a tree similarity measure model that can construct this kind of mapping. The similarity measure model takes into account all the information on nodes' concepts, weights, and values. The conceptual similarity and the value similarity between two trees are evaluated based on the constructed mapping, and the final similarity measure is assessed as a weighted sum of their conceptual and value similarities. The effectiveness of the proposed similarity measure model is shown by an illustrative example and is also demonstrated by applying it into a recommender system.

Keywords

Tree similarity measure, tree mapping, one-to-many mapping, tree structured business data

INTRODUCTION

Tree structured data are becoming ubiquitous nowadays in many applications. They are widely used to represent information in computational biology (Ouangraoua and Ferraro 2009), ontology management (Born et al. 2008; Solskinnsbakk et al. 2012; Xue et al. 2009), case based reasoning (Ricci and Senter 1998), document classification (Lin et al. 2008), e-business applications (Bhavsar et al. 2004; Yang et al. 2005), complex product and user profile representation (Wu et al. 2010; Wu and Zhang 2011), and so on. Evaluating the data similarity is usually an essential part of these applications. For example, in case based reasoning, a key is to search for the most similar cases to a new problem. As ontology usage becomes more prevalent in e-business decision support systems, it is essential to assess the similarity between tree structured ontologies (Zhao et al. 2012). In recommender systems in an e-business environment, it is important to find the similar users or products. Therefore, an effective similarity measure for tree structured data is needed in the above situations.

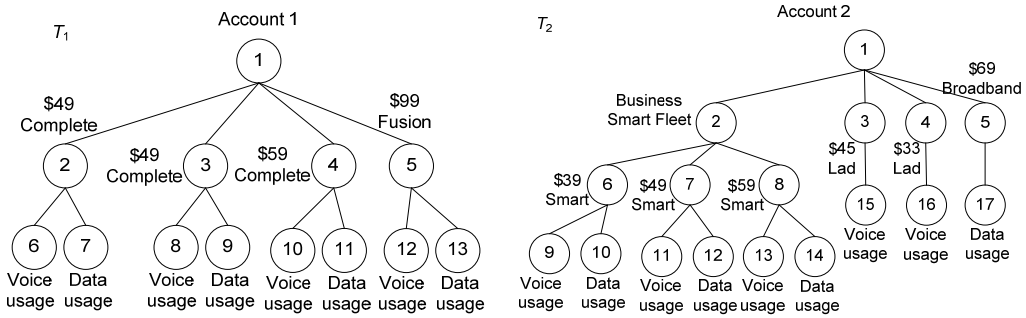

Figure 1: Two tree structured business user profiles in the telecom industry

Figure 1 shows the usage data structures of two business users in the telecom industry. Using this as an example, a business user account is comprised of several services, each service is associated with a plan, and each plan provides several specific service items, which constructs a tree structure. The nodes in the tree are assigned with attributes, such as plan names, plan family names, service item names and so on, to express their semantic meanings. The tree structures reflect the semantic relations between these attributes. Taking T_2 as an example, the three smart mobile plans represented by nodes 6, 7 and 8 construct a fleet, in which the services can share

their included values. To express the fleet relation, they are under one sub-tree. The tree is also assigned with values. For a specific user, his/her usage amounts or costs of service items should be assigned to the relevant attributes. Different nodes/attributes may also have different importance degrees in real applications. Therefore, the tree structures, nodes' attributes, weights and values of different trees are probably all different. This kind of complex tree structured business data is our focus in this research. To compare two such tree structured data, the tree structures, concepts of nodes and the values should all be considered. An issue of the attribute concept is that one concept intension may include several other concepts intensions. For example, the *Fusion* product provides the service items of both *Lad* and *Broadband*. The concept intension inclusion relations must be considered in the tree similarity measure.

The research on the similarity measure models of tree structured data has attracted a great deal of attention from many application fields. In previous research, trees are compared from both the structural and semantic aspects. Tree edit distance model (Bille 2005) is the most widely used method to compare the structures of ordered or unordered, labelled trees. The model measures the degree of similarity between two trees by the minimum cost of the edit operation sequences that convert one tree into another. The edit operations give rise to an edit distance mapping, which is a graphical specification of what edit operations apply to each node in the two labelled trees (Zhang 1993). Considering structural constraints, constrained edit distance (Zhang 1996) requires that disjoint sub-trees be mapped to disjoint sub-trees. Less constrained edit distance (Lu et al. 2001) loosens the constraint and allows one sub-tree of T_1 to be mapped to more than one sub-tree of T_2, which constructs a one-to-many mapping. As tree structures reflect the semantic meanings of the objects, these structural constraints are necessary in many applications. The semantic or conceptually similarity between attributes is also taken into account when comparing two trees (Ricci and Senter 1998; Xue et al. 2009). Only conceptual similar attributes can be mapped or transformed. To match and compare ontologies in e-business decision support systems, Zhao et al. (2012) developed an algorithm that combines syntactic analysis measuring the difference between tokens by the edit distance, semantic analysis based on WordNet as semantic relation and similarity assessment of tree-structured graphs with the Tversky similarity model. In a business environment, the data are more complex. Besides tree structures and attributes concepts, nodes' values and weights are also considered (Wu et al. 2010; Wu et al. 2011; Wu and Zhang 2011), and a comprehensive similarity measure model is developed. However, in that similarity measure model, only one-to-one mapping between two trees is constructed, i.e., one sub-tree can be mapped to only one sub-tree in the other tree, which does not deal with the concept intension inclusion problem illustrated in Figure 1. In the example, the sub-tree rooted at *Fusion* should be mapped to the sub-trees rooted at *Lad* and *Broadband* at the same time. The less constrained edit distance mapping (Lu et al. 2001) is suitable in this situation. In this research, a comprehensive similarity measure on tree structured business data that can deal with the concept intension inclusion issue will be developed.

The contribution of this paper is that a comprehensive similarity measure on tree structured business data is proposed. The similarity measure fully considers the tree structures, nodes' concepts, values and weights. To calculate the similarity between two trees, the concept corresponding parts are identified first by constructing a maximum conceptual similarity tree mapping. To deal with the concept intension inclusion issues, a one-to-many mapping is constructed. The conceptual and value similarity between two trees are then computed separately, and the final similarity is assessed as the weighted sum of their conceptual and value similarities.

This paper is organised as follows. Section 2 formally describes the features of tree structured data. A comprehensive similarity measure model for tree structured data is provided in Section 3. To show the effectiveness of the proposed tree similarity measure, an illustrative example to compare two tree structured business data is given in Section 4. In Section 5, the similarity measure model is applied to a recommender system to prove its effectiveness. Finally, conclusions and future study are discussed in Section 6.

TREE STRUCTURED DATA DEFINITION

A tree is defined as a directed graph $T=(V, E)$, where the underlying undirected graph has no cycles and there is a distinguished root node in V , denoted by $root(T)$, so that for all nodes $v \in V$, there is a path in T from $root(T)$ to node v (Valiente 2002).

In real applications, the definition is usually extended to represent practical objects. In our research, a tree structured data model for business data is proposed by adding the following features to the definition.

- Nodes in a tree are assigned semantic meanings. A domain attribute term set A, which is a set of symbols to specify semantic meanings to nodes, is introduced. There exists an attribute assignment function $a: V \rightarrow A$ so that each node in the tree is assigned an attribute. The attribute terms can be divided into basic attributes and complex attributes. The complex attribute represents the semantic concept combined with several basic attributes. The basic attribute is a unary variable.

- An attribute conceptual similarity measure within the domain attribute term set *A* is defined as a set of mappings *sc*:*A*×*A*→[0,1], in which each mapping denotes the conceptual similarity between two attributes (Xue et al. 2009). For any $a_1, a_2 \in A$, we say a_1 and a_2 are similar if $sc(a_1, a_2) > \varepsilon$, where ε is the threshold of the similar relation. The larger $sc(a_1, a_2)$ is, the more similar the two attributes are. The conceptual similarity measures can be given by domain experts or inferred from the domain ontology that describes the relations between the attributes.
- An attribute conceptual inclusion relation within the domain attribute term set *A* is introduced as a set $R \subset A \times A$. For any $(a_1, a_2) \in R$, the intension of the concept of a_1 includes the intension of the concept of a_2.
- For each basic attribute $b \in A$, it is associated with a value domain D_b and a value similarity measure $s_b : D_b \times D_b \to [0,1]$.
- A weight function *w*:*V*→[0,1] assigns a weight to each node to represent its importance degree to its siblings.

A COMPREHENSIVE SIMILARITY MEASURE MODEL ON TREE STRUCTURED DATA WITH NODE CONCEPT INCLUSION RELATIONS

To evaluate the similarity between two trees, both the concepts and values of nodes need to be compared. The conceptual similarity and value similarity between two trees are defined respectively, and the final similarity measure between them is assessed as the weighted sum of their conceptual and value similarities.

In the following, the symbols in (Zhang 1996) are used to represent trees and nodes. Suppose that we have a numbering for each tree. Let *t*[*i*] be the *i*th node of the tree *T* in the given numbering. Let *T*[*i*] be the sub-tree rooted at *t*[*i*] and *F*[*i*] be the unordered forest obtained by deleting *t*[*i*] from *T*[*i*]. Let $t_1[i_1]$, $t_1[i_2]$, ..., $t_1[i_{n_i}]$ be the children of $t_1[i]$ and $t_2[j_1]$, $t_2[j_2]$, ..., $t_2[j_{n_j}]$ be the children of $t_2[j]$.

Conceptual similarity between two trees

Definition 1: Conceptual similarity. Let S_T be the set of trees to be compared and S_F be the set of forests derived from S_T. A conceptual similarity between two trees is defined as a set of mappings $sc_T : S_T \times S_T \to [0,1]$, in which each mapping denotes the conceptual similarity between the corresponding two trees. A conceptual similarity between two forests is defined as a set of mappings $sc_F : S_F \times S_F \to [0,1]$, in which each mapping denotes the conceptual similarity between the corresponding two forests.

The conceptual similarity has the following features.

- sc_T and sc_F are symmetric, i.e. for any $T_1[i]$, $T_2[j]$, $sc_T(T_1[i], T_2[j]) = sc_T(T_2[j], T_1[i])$, $sc_F(F_1[i], F_2[j]) = sc_F(F_2[j], F_1[i])$.
- For two completely same trees/forests, their conceptual similarity value reaches the maximum, i.e. $sc_T(T_1[i], T_1[i]) = 1$, $sc_F(F_1[i], F_1[i]) = 1$.
- Let ϕ represent the empty tree or forest. To compare a tree or forest with ϕ, the similarity contribution is assumed to be 0, i.e. $sc_T(T_1[i], \phi) = 0$, $sc_F(F_1[i], \phi) = 0$.

The concept of a tree is derived from the concepts of nodes' attributes and tree structures. Both of these two aspects should be considered when evaluating the conceptual similarity between two trees. To compare two trees, their concept corresponding parts should be identified first; the corresponding pairs are then compared separately, and finally aggregated into one value. When determining the corresponding node pairs, both the structural and conceptual constraints should be satisfied. For the structural constraints, the tree edit distance mapping (Bille 2005) is introduced. In particular, a less constrained tree edit distance mapping (Lu et al. 2001) is constructed to solve the concept intension inclusion issues. For the conceptual constraints, only nodes with conceptual similar attributes are allowed to be matched.

Tree conceptual similarity calculation formula

Given two trees $T_1[i]$ and $T_2[j]$ to be compared, according to the matching situations of their roots $t_1[i]$ and $t_2[j]$, three cases are considered: $t_1[i]$ and $t_2[j]$ are matched; $t_1[i]$ is matched to $t_2[j]$'s children; $t_2[j]$ is matched to $t_1[i]$'s children. The matching situation with the maximum conceptual similarity value is the best matching.

In the first case, $t_1[i]$ and $t_2[j]$ are matched. The conceptual similarity between $T_1[i]$ and $T_2[j]$ is calculated as:

$$sc_{T_1}(T_1[i],T_2[j]) = \begin{cases} sc(a(t_1[i]),a(t_2[j])), & F_1[i]=\phi, F_2[j]=\phi \\ \alpha \cdot sc(a(t_1[i]),a(t_2[j])) + (1-\alpha)\cdot \sum_{t=1}^{n_j} w_{j_t} \cdot sc_{T_1}(T_1[i],T_2[j_t]), & F_1[i]=\phi, F_2[j]\neq\phi \\ \alpha \cdot sc(a(t_1[i]),a(t_2[j])) + (1-\alpha)\cdot \sum_{t=1}^{n_i} w_{i_t} \cdot sc_{T_1}(T_1[i_t],T_2[j]), & F_1[i]\neq\phi, F_2[j]=\phi \\ \alpha \cdot sc(a(t_1[i]),a(t_2[j])) + (1-\alpha)\cdot sc_F(F_1[i],F_2[j]), & F_1[i]\neq\phi, F_2[j]\neq\phi \end{cases} \quad (1)$$

where $a(t_1[i])$ and $a(t_2[j])$ represent the attributes of $t_1[i]$ and $t_2[j]$ respectively, w_{j_t} and w_{i_t} are the weights of $t_2[j_t]$ and $t_1[i_t]$ respectively, and α is the influence factor of the parent node. According to the conditions, whether $t_1[i]$ and $t_2[j]$ are leaves, four situations are listed in Formula (1). In the first situation, $t_1[i]$ and $t_2[j]$ are both leaves, their conceptual similarity is equivalent to the conceptual similarity of their attributes. In the second and third situations, one node is a leaf and the other is an inner node. As the concept of a tree is dependent not only on its root's attribute, but also on its children's, the children of the inner node are also considered in the formulas. In the last situation, both $t_1[i]$ and $t_2[j]$ have children. Their children construct two forests $F_1[i]$ and $F_2[j]$, which are compared with the forest similarity measure $sc_F(F_1[i],F_2[j])$.

Taking Figure 1 as an example, for node 6 in T_1 and node 9 in T_2, $sc_{T_1}(T_1[6],T_2[9])$ = *sc*("*Voice usage*", "*Voice usage*"). For node 2 in T_1 and node 6 in T_2, $sc_{T_1}(T_1[2],T_2[6]) = \alpha \cdot sc$("\$49 *Complete*", "\$39 *Smart*")+ $(1-\alpha)\cdot sc_F(F_1[2],F_2[6])$.

In the second case, $t_1[i]$ is matched to $t_2[j]$'s children. In this case, the concept level of $t_1[i]$ is lower than that of $t_2[j]$. $T_1[i]$ is mapped to the sub-tree of $T_2[j]$, which has the maximum conceptual similarity with $T_1[i]$. The conceptual similarity between $T_1[i]$ and $T_2[j]$ is represented as:

$$sc_{T_2}(T_1[i],T_2[j]) = \max_{1\le t\le n_j}\{0.5\cdot(1+w(t_2[j_t]))\cdot sc_T(T_1[i],T_2[j_t])\} \quad (2)$$

For example, Figure 2 shows the usage record structure of a business user in the telecom industry. The customer has only mobile services. When comparing T_3 in Figure 2 with T_2 in Figure 1, node 1 in T_3 is probably matched to node 2 in T_2.

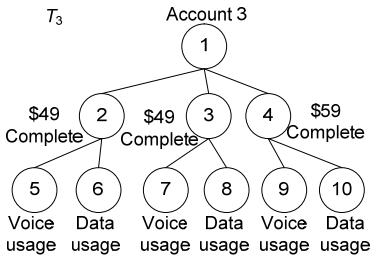

Figure 2: A tree structured business user profile in the telecom industry

The third situation is similar to the second situation. The conceptual similarity between $T_1[i]$ and $T_2[j]$ is calculated as:

$$sc_{T_3}(T_1[i], T_2[j]) = \max_{1 \le t \le n_i} \{0.5 \cdot (1 + w(t_1[i_t])) \cdot sc_T(T_1[i_t], T_2[j])\} \quad (3)$$

Considering the three cases listed above, the conceptual similarity between $T_1[i]$ and $T_2[j]$ is:

$$sc_T(T_1[i], T_2[j]) = \max\{sc_{T_1}, sc_{T_2}, sc_{T_3}\} \quad (4)$$

Forest conceptual similarity calculation formula

During the computation process of the conceptual similarity between two trees, the conceptual similarity between two forests is used. In the following, the calculation method of $sc_F(F_1[i], F_2[j])$ is given. The concept corresponding sub-trees are first identified based on both their concepts and structures, and then compared separately. Finally, these local similarity measures are weighted aggregated. The whole process can be divided into three steps.

Step 1: divide the forests into several conceptually similar forest groups.

As mentioned before, only conceptually similar nodes can be matched and compared. In this step, the conceptual similar nodes are grouped. The roots of $F_1[i]$ and $F_2[j]$ construct a bipartite graph, G_{F_1,F_2}=(*V*, *E*), in which V=$\{t_1[i_1], t_1[i_2], ..., t_1[i_{n_i}]\} \cup \{t_2[j_1], t_2[j_2], ..., t_2[j_{n_j}]\}$, $E = \{(t_1[i_p], t_2[j_q]) \mid sc_T(T_1[i_p], T_2[j_q]) > \varepsilon\}$. G_{F_1,F_2} can be divided into several disconnected sub-graphs $G_1, G_2, \ldots, G_g$. Each sub-graph, G_t, represents a conceptual similar forest pair ($F_{1,t}[i]$, $F_{2,t}[j]$).

Step 2: calculate the conceptual similarity of each conceptually similar forest pair

The conceptual similarity of a conceptually similar forest pair $(F_{1,t}[i], F_{2,t}[j])$ is denoted as $sc_{Fs}(F_{1,t}[i], F_{2,t}[j])$, in which $F_{1,t}[i]$=$\{T_1[i_1], T_1[i_2], \ldots, T_1[i_{t_i}]\}$, $F_{2,t}[j]$=$\{T_2[j_1], T_2[j_2], \ldots, T_2[j_{t_j}]\}$.

The key point for comparing two forests is to construct a mapping to identify the most conceptually corresponding tree pairs. The mapping should be satisfied with both conceptual and structural constraints, as mentioned before. The mapping of trees in a conceptually similar forest pair obviously satisfies the conceptual constraints. For the structural constraints, a one-to-one mapping or a one-to-many mapping that is used to deal with the concept inclusion needs to be constructed, i.e., a less constrained tree edit distance mapping (Lu et al. 2001) needs to be constructed. According to the mapping types, two cases are considered separately.

In the first case, a one-to-one mapping between $F_{1,t}[i]$ and $F_{2,t}[j]$ is constructed. The sub-bipartite graph is denoted as $G_t = (V_{1,t} \cup V_{2,t}, E_t)$, in which $V_{1,t}$=$\{t_1[i_1], t_1[i_2], ..., t_1[i_{t_i}]\}$, $V_{2,t}$=$\{t_2[j_1], t_2[j_2], ..., t_2[j_{t_j}]\}$. For any $t_1[i_p] \in V_{1,t}$ and $t_2[j_q] \in V_{2,t}$, a weight is assigned to edge ($t_1[i_p]$, $t_2[j_q]$) as $w_{p,q} = sc_T(T_1[i_p], T_2[j_q])$. A maximum weighted bipartite matching (Jungnickel 2008) of G_t, $MWBM_{F_{1,t},F_{2,t}}$ is constructed. The conceptual similarity between $F_{1,t}[i]$ and $F_{2,t}[j]$ in this case is calculated as

$$sc_{Fs_1}(F_{1,t}[i], F_{2,t}[j]) = \sum_{(t_1[k], t_2[l]) \in MWBM_{F_{1,t},F_{2,t}}} w_{k,l} \cdot sc_T(T_1[k], T_2[l]), \quad (5)$$

where $w_{k,l} = (w(t_1[k]) + w(t_2[l]))/2$.

In the second case, if concept inclusion relations exist between nodes in $V_{1,t}$ and nodes in $V_{2,t}$, a one-to-many mapping between $F_{1,t}[i]$ and $F_{2,t}[j]$ will be constructed. It is assumed that there is a domain ontology that represents the inclusion relations between attribute terms. All the inclusion relations between $V_{1,t}$ and $V_{2,t}$ are identified first, constructing a concept inclusion relation set *IR*. Each relation in *IR* is a binary tuple (*t*[*n*], *S*), in which *t*[*n*] is a node, and *S* are the corresponding nodes whose concepts are included in *t*[*n*]. To construct the *IR* effectively and efficiently, domain knowledge, such as business rules, should be introduced. Taking trees in Figure 1 as examples, the domain ontology will define the inclusion relation between *Fusion* and *Lad*, *Broadband*. The domain knowledge will indicate that the *Fusion* product is suitable for a heavy cost user, i.e.

Fusion should be matched to the *Lad* and *Broadband* with the biggest cost values. With the above knowledge, the concept inclusion relation between the tree nodes can be identified.

Considering an inclusion relation r=(t[n],S), let $F_{1-r,t}[i]$, $F_{2-r,t}[j]$ be forests obtained by removing nodes in the relation r. The conceptual similarity between $F_{1,t}[i]$ and $F_{2,t}[j]$ in this case is calculated as the sum of conceptual similarity between T[n] and the forest rooted at S, F_S, and the conceptual similarity between $F_{1-r,t}[i]$ and $F_{2-r,t}[j]$.

Let $sc_R(r)$ represent the conceptual similarity between T[n] and F_S. According to different mapping situations between T[n] and F_S, $sc_R(r)$ is calculated in different ways. If t[n] is a leaf node, $sc_R(r)$ is calculated as a weighted sum of the similarities between T[n] and the trees in F_S. Otherwise, t[n]'s children must be considered. t[n]'s children can be mapped to the nodes in S or the children of S. $sc_R(r)$ is calculated as follows,

$$sc_R(r) = \begin{cases} w_{n,S} \cdot \sum_{t_S[j_p] \in S} w_{j_p} \cdot sc_T(T[n], T_S[j_p]), & F_n = \phi \\ w_{n,S} \cdot (1-\alpha) \cdot sc_{F_S}(F[n], F_S), & t[n]'s\ children\ are\ similar\ to\ S \\ w_{n,S} \cdot (\alpha \cdot sc_a(t[n], S) + (1-\alpha) \cdot sc_{Fs1}(F[n], F_{SC})), & t[n]'s\ children\ are\ not\ similar\ to\ S \end{cases} \quad (6)$$

where $w_{n,S} = (w(t[n]) + \sum_{t_S[j_p] \in S} w(t_S[j_p]))/2$, $w_{j_p} = w(t_S[j_p]) / \sum_{t_S[j_p] \in S} w(t_S[j_p])$, α is the influence factor of the parent, F_{SC} represents the children of nodes in S, $sc_a(t[n], S)$ is the concept similarity between t[n] and S.

$$sc_a(t[n], S) = \sum_{t_S[j_p] \in S} w_{j_p} \cdot sc(a(t[n]), a(t_S[j_p])), \quad (7)$$

where $w_{j_p} = w(t_S[j_p]) / \sum_{t_S[j_p] \in S} w(t_S[j_p])$. In the above formula, to satisfy the structural constraints, only one-to-one mapping between F[n] and F_{SC} can be constructed. The roots' weights of F_{SC} should also be normalised.

There are usually more than one inclusion relations. The conceptual similarity between $F_{1,t}[i]$ and $F_{2,t}[j]$ for the second case is calculated as

$$sc_{Fs_2}(F_{1,t}[i], F_{2,t}[j]) = \max_{r \in IR} \{sc_R(r) + sc_{Fs}(F_{1-r,t}[i], F_{2-r,t}[j])\}. \quad (8)$$

Finally, the conceptual similarity of a conceptual similar forest pair ($F_{1,t}[i]$, $F_{2,t}[j]$) is the maximum of the two cases discussed above:

$$sc_{Fs}(F_{1,t}[i], F_{2,t}[j]) = \max\{sc_{Fs_1}, sc_{Fs_2}\}. \quad (9)$$

Step 3: aggregate the conceptual similarity of each conceptual similar forest pair

$$sc_F(F_1[i], F_2[j]) = \sum_{t=1}^{g} sc_{Fs}(F_{1,t}[i], F_{2,t}[j]). \quad (10)$$

Maximum conceptual similarity tree mapping

During the computation process of the conceptual similarity between two trees, the maximum conceptual similarity tree mapping is constructed to identify the most conceptual corresponding node pairs. It includes two types, one-to-one and one-to-many mappings. The one-to-many mapping indicates the concept inclusion relations.

Based on the mapping, nodes in two trees can be divided into three kinds: conceptual corresponding nodes, semi-conceptual corresponding nodes, and not corresponding nodes. The conceptual corresponding nodes are the nodes that appear in the maximum conceptual similarity tree mapping. The semi-conceptual corresponding nodes are the nodes that do not appear in the mapping but their decedents appear in the mapping. Not corresponding

nodes are the nodes that neither themselves nor their decedents appear in the mapping. Obviously, the roots of two trees to be compared must be a conceptual corresponding or semi-conceptual corresponding node pair.

Value similarity between two trees

For a specific tree structured object, some nodes are assigned values to describe the degrees of the relevant attributes. For each branch of a tree, as nodes in the branch represent a common concept at different levels, only one node, which is usually the leaf, is assigned a value. Besides the concepts, the values of two trees should also be compared to comprehensively evaluate their similarity measure.

Let $v(t[i])$ represent the value of node $t[i]$. $v(t[i])=null$ if $t[i]$ is not assigned a value. Given two trees $T_1[i]$ and $T_2[j]$ to be compared, a maximum conceptual similarity tree mapping M_s has been constructed. According to different situations whether $t_1[i]$ and $t_2[j]$ are assigned values or not, the value similarity between $T_1[i]$ and $T_2[j]$, $sv_T(T_1[i],T_2[j])$, is calculated in the following three cases.

Case1: both the two roots are assigned values, i.e. $v(t_1[i]) \neq null$, $v(t_2[j]) \neq null$

$$sv_T(T_1[i],T_2[j]) = s_a(v(t_1[i]),v(t_2[j])). \tag{11}$$

Case2: only one root is assigned a value, i.e. $v(t_1[i]) \neq null$, $v(t_2[j]) = null$

In this case, the values of $t_2[j]$'s sub-trees, which are corresponding to $T_1[i]$, are aggregated and compared with $v(t_1[i])$.

$$sv_T(T_1[i],T_2[j]) = s_a(v(t_1[i]),v(T_2[j])), \tag{12}$$

where $v(T_2[j]) = \sum_{(t_1[i_p],t_2[j_t])\in M_s} v(T_2[j_t])$ represents the aggregated value of the corresponding nodes in $T_2[j]$.

Case3: both the two roots are not assigned values, i.e. $v(t_1[i]) = null$, $v(t_2[j]) = null$

In this case, the values of their corresponding sub-trees are compared. Based on the maximum conceptual similarity tree mapping M_s, only conceptual corresponding or semi-conceptual corresponding node pairs are compared. The semi-conceptual corresponding nodes are replaced with their nearest decedents that are in the mapping M_s. The weights of the replaced nodes are adjusted accordingly. If there exists one-to-many mapping, replace the "many" part with one node, set the value of the new node as the sum of the original nodes' values, and set the weight of the new node as the sum of the original nodes' weights. A new mapping M_s' between $t_1[i]$ and $t_2[j]$' children will then be constructed.

$$sv_T(T_1[i],T_2[j]) = \sum_{(t_1[i_p],t_2[j_q])\in M_s'} w_{p,q} \cdot sv_T(T_1[i_p],T_2[j_q]), \tag{13}$$

where $w_{p,q} = (w(t_1[p]) + w(t_2[q]))/2$.

Similarity measure between two trees

Based on the conceptual similarity and value similarity of two trees, the final comprehensive similarity measure between T_1 and T_2 is defined as follows:

$$sim(T_1,T_2) = \alpha_1 \cdot sc_T(T_1,T_2) + \alpha_2 \cdot sv_T(T_1,T_2), \tag{14}$$

where $\alpha_1 + \alpha_2 = 1$.

AN ILLUSTRATIVE EXAMPLE TO COMPARE TWO TREE STRUCTURED BUSINESS DATA

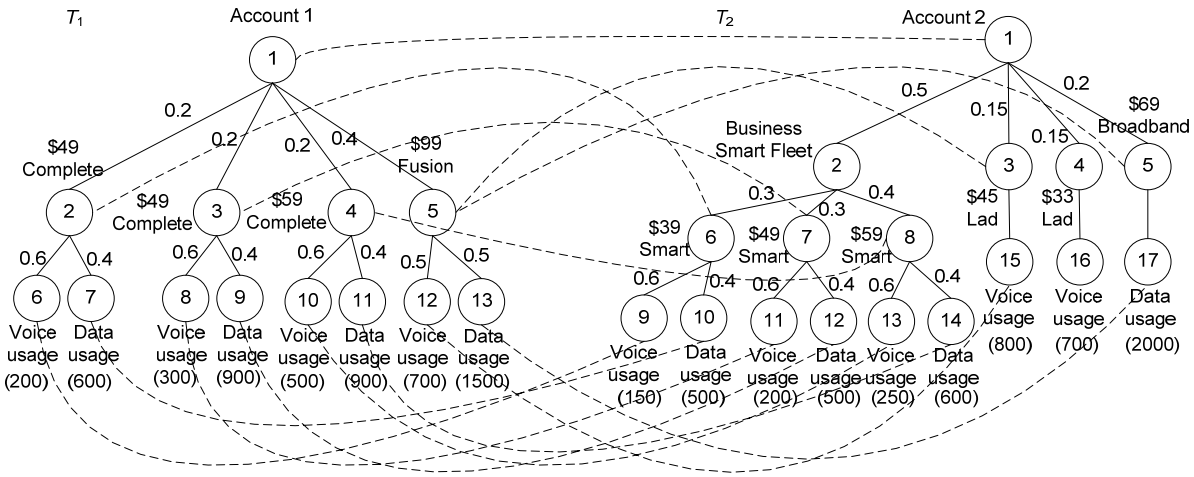

Figure 3: Two tree structured business user profiles in the telecom industry

To show the effectiveness of the proposed tree similarity measure, an illustrative example to compare two tree structured business data in Figure 1 is given in this section. The trees are assigned with values and weights, which are shown in Figure 3. The number beside the edge is the weight of the child. The number under each leaf represents its value. The attribute conceptual similarity measure is defined as: *sc*(*$49Complete*,*$59Smart*)=0.7, *sc*(*$49Complete*,*$49Smart*)=0.8, *sc*(*$49Complete*,*$39Smart*)=0.7, *sc*(*$59Complete*,*$59Smart*)=0.8, *sc*(*$59Complete*,*$49Smart*)=0.7, *sc*(*$59Complete*,*$39Smart*)=0.6, *sc*(*$99Fusion*,*$69Broadband*)=0.6, *sc*(*$99Fusion*,*$45Lad*)=0.5, *sc*(*$99Fusion*,*$33Lad*)=0.4. Two kinds of attribute conceptual inclusion relations are defined: the *Fusion* includes one *Lad* and one *Broadband*, and *Business Smart Fleet* includes *Mobiles*. Let the influence factor of the parent node α =0.5, the conceptual similarity between T_1 and T_2 is calculated as 0.77 by the proposed tree similarity measure. A maximum conceptual similarity tree mapping is constructed which is described with the dashed lines in Figure 3. It can be seen from the mapping that node 5 in T_1 is mapped to nodes 3 and 5 in T_2 at the same time, and sub-trees under nodes 2, 3, 4 in T_1 are mapped to the sub-tree under node 2 in T_2, which solves the concept intension inclusion problem. Based on the maximum conceptual similarity tree mapping, the values of two trees are compared and the value similarity is calculated as 0.52. The final similarity between T_1 and T_2 can be calculated as the weighted sum of their conceptual and value similarity by Formula (14).

APPLICATION OF THE TREE SIMILARITY MEASURE IN RECOMMENDER SYSTEMS

An application of the proposed tree similarity measure in a recommender system is shown in this section. Recommender systems (Adomavicius and Tuzhilin 2005) are widely used nowadays in the e-business environment. It is important to find the similar users or items to make recommendations. For example, the content based recommendation technique recommends items that are similar to the ones preferred before by a specific user (Pazzani and Billsus 2007). In many application fields, such as the telecom industry, items are so complex that they need to be represented as tree structures. The tree similarity measure developed above is suitable in these applications.

In content based recommender systems, the predicted rating of user *u* to item *i* is calculated as

$$P_{u,i} = \frac{\sum_{n \in I_u} r_{u,n} \times sim(i,n)}{\sum_{n \in I_u} sim(i,n)}, \tag{15}$$

where I_u is the item set user *u* has rated, $r_{u,n}$ is the rating of user *u* to item *n*, $sim(i,n)$ represents the similarity between item *n* and item *i*. Here, the tree similarity measure is used to calculate the similarity between tree structured items.

To show the effectiveness of the similarity measure, the HetRec 2011 MovieLens Data Set (HetRec 2011, http://ir.ii.uam.es/hetrec2011) is used to test the recommendation results. The dataset is an extension of MovieLens10M dataset (http://www.grouplens.org/node/73), which contains personal ratings and tags about movies. From the original dataset, only those users with both ratings and tags have been maintained. In the dataset, the movies are linked to Internet Movie Database (IMDb) and RottenTomatoes (RT) movie review systems. Each movie does have its IMDb and RT identifiers, English and Spanish titles, picture URLs, genres, directors, actors (ordered by "popularity"), RT audience' and experts' ratings and scores, countries, and filming locations. In our experiment, each movie is represented as a tree structured object. The structure is shown in Figure 4. There are 2113 users in the data set. In our experiment, the latest 50 ratings of each user are used, and the latest 10 ratings of each user are used as test sets to calculate the mean absolute error (MAE).

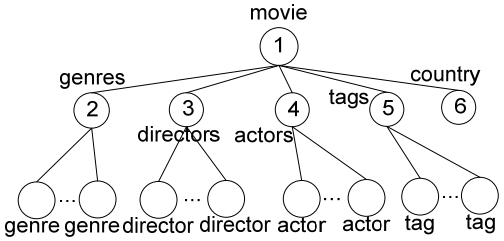

Figure 4: The movie tree structure

The overall MAE of the content based recommendation approach with Formula (15) by use of our proposed similarity measure is 0.703. For the sake of contrast, the collaborative filtering (CF) recommendation approach (Schafer et al. 2007) is also used in the experiment. The overall MAE of CF approach is 0.71. As CF has the new item problem, 523 test cases cannot be recommended, while they can be recommended using the content based approach and their MAE is 0.679. For the items that are not rated by sufficient amounts of users, the performance of CF is not guaranteed. However, as the semantic features of the movies are fully considered, the recommendation performance is not influenced in our approach. For example, for the items that are rated less than five times, the MAE of CF is 0.902 while the MAE of our approach is 0.71.

From the experiment result, it can be seen that the proposed tree similarity measure can effectively support the recommender system to make recommendations for tree structured items.

CONCLUSION

This paper proposes a similarity measure on tree structured business data, which fully considers the tree structures, nodes' concepts, values and weights. To calculate the similarity between two trees, the concept corresponding parts are identified first by constructing a maximum conceptual similarity tree mapping. To deal with the concept intension inclusion issues, a one-to-many mapping is constructed. The conceptual and value similarity between two trees are then computed separately, and the final similarity is assessed as the weighted sum of their conceptual and value similarities. An illustrative example shows that the proposed tree similarity measure can effectively solve the concept intension inclusion problem. The proposed tree similarity measure is then applied to a recommender system, and the experiment is carried out to evaluate the performance. The experiment result shows that the proposed tree similarity measure does effectively support the recommender system to make recommendations for tree structured items. In the future, the proposed tree similarity measure will be used in a real recommender system for business users in the telecom industry and the similarity measure will be improved and evaluated in real applications.

REFERENCES

Adomavicius, G., and Tuzhilin, A. 2005. "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), pp 734-749.

Bhavsar, V.C., Boley, H., and Yang, L. 2004. "A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in E-Business Environments," *Computational Intelligence* (20:4), pp 584-602.

Bille, P. 2005. "A Survey on Tree Edit Distance and Related Problems," *Theoretical computer science* (337:1-3), pp 217-239.

Born, M., Filipowska, A., Kaczmarek, M., Markovic, I., Starzecka, M., and Walczak, A. 2008. "Business Functions Ontology and Its Application in Semantic Business Process Modelling," *19th Australasian Conference on Information Systems (ACIS 2008)*, Christchurch, pp. 136-145.

Jungnickel, D. 2008. *Graphs, Networks, and Algorithms*. Springer Verlag, pp 419-430.

Lin, Z., Wang, H., McClean, S., and Liu, C. 2008. "All Common Embedded Subtrees for Measuring Tree Similarity," *International Symposium on Computational Intelligence and Design*, pp. 29-32.

Lu, C., Su, Z.-Y., and Tang, C. 2001. "A New Measure of Edit Distance between Labeled Trees," in: *Computing and Combinatorics,* J. Wang (ed.). Springer Berlin / Heidelberg, pp. 338-348.

Ouangraoua, A., and Ferraro, P. 2009. "A Constrained Edit Distance Algorithm between Semi-Ordered Trees," *Theoretical computer science* (410:8-10), pp 837-846.

Pazzani, M., and Billsus, D. 2007. "Content-Based Recommendation Systems," in: *The Adaptive Web,* P. Brusilovsky, A. Kobsa and W. Nejdl (eds.). Springer Berlin / Heidelberg, pp. 325-341.

Ricci, F., and Senter, L. 1998. "Structured Cases, Trees and Efficient Retrieval," *Advances in Case-Based Reasoning* (1488), pp 88-99.

Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. 2007. "Collaborative Filtering Recommender Systems," in: *The Adaptive Web,* P. Brusilovsky, A. Kobsa and W. Nejdl (eds.). Springer Berlin / Heidelberg, pp. 291-324.

Solskinnsbakk, G., Gulla, J.A., Haderlein, V., Myrseth, P., and Cerrato, O. 2012. "Quality of Hierarchies in Ontologies and Folksonomies," *Data & Knowledge Engineering* (74:0), pp 13-25.

Valiente, G. 2002. *Algorithms on Trees and Graphs*. New York : Springer, pp 16-22.

Wu, D., Lu, J., and Zhang, G. 2010. "A Hybrid Recommendation Approach for Hierarchical Items," *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, pp. 492-497.

Wu, D., Lu, J., and Zhang, G. 2011. "Similarity Measure Models and Algorithms for Hierarchical Cases," *Expert Systems with Applications* (38:12), pp 15049-15056.

Wu, D., and Zhang, G. 2011. "Fuzzy Similarity Measure Model for Trees with Duplicated Attributes," in: *Nonlinear Mathematics for Uncertainty and Its Applications,* S. Li, X. Wang, Y. Okazaki, J. Kawabe, T. Murofushi and L. Guan (eds.). Springer Berlin / Heidelberg, pp. 333-340.

Xue, Y., Wang, C., Ghenniwa, H., and Shen, W. 2009. "A Tree Similarity Measuring Method and Its Application to Ontology Comparison," *Journal of Universal Computer Science* (15:9), pp 1766-1781.

Yang, L., Sarker, B., Bhavsar, V., and Boley, H. 2005. "A Weighted-Tree Simplicity Algorithm for Similarity Matching of Partial Product Descriptions," *Proceedings of The International Society for Computers and Their Applications (ISCA) 14th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE-2005)*, Toronto, Ontario, Canada, pp. 55-60.

Zhang, K. 1993. "A New Editing Based Distance between Unordered Labeled Trees," in: *Combinatorial Pattern Matching,* A. Apostolico, M. Crochemore, Z. Galil and U. Manber (eds.). Springer Berlin / Heidelberg, pp. 254-265.

Zhang, K. 1996. "A Constrained Edit Distance between Unordered Labeled Trees," *Algorithmica* (15:3), pp 205-222.

Zhao, Y., Li, Z., Wang, X., and Halang, W.A. 2012. "Decision Support in E-Business Based on Assessing Similarities between Ontologies," *Knowledge-Based Systems* (32:0), pp 47-55.

COPYRIGHT

Dianshuang Wu, Guangquan Zhang, Jie Lu, Wolfgang A. Halang © 2012. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.