

User Model Clustering

Loc Nguyen

University of Science, Ho Chi Minh, Vietnam
Email: ng_phloc@yahoo.com

Received 15 March 2014; revised 17 April 2014; accepted 4 May 2014

Copyright © 2014 by author and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

User model which is the representation of information about user is the heart of adaptive systems. It helps adaptive systems to perform adaptation tasks. There are two kinds of adaptations: 1) Individual adaptation regarding to each user; 2) Group adaptation focusing on group of users. To support group adaptation, the basic problem which needs to be solved is how to create user groups. This relates to clustering techniques so as to cluster user models because a group is considered as a cluster of similar user models. In this paper we discuss two clustering algorithms: *k*-means and *k*-medoids and also propose dissimilarity measures and similarity measures which are applied into different structures (forms) of user models like vector, overlay, and Bayesian network.

Keywords

User Model, Cluster

1. Introduction

User model is the representation of personal traits (or characteristics) about user, such as demographic information, knowledge, learning style, goal, interest, etc. Learner model is defined as user model in learning context in which user is a learner who profits from (adaptive) learning system. We note that learner model, student model, and user model are the same terms in learning context. Adaptive systems exploit valuable information in user model so as to provide adaptation effects, *i.e.*, to behave different users in different ways. For example, the adaptive systems tune learning materials to a user in order to provide the best materials to her/him. The usual adaptation effect is to give individually adaptation to each user, but there is a demand to provide adaptation to a group or community of users. Consequently, all users in the same group will profit from the same learning materials, teaching methods, etc. because they have the common characteristics. So there are two kinds of adaptations:

- 1) *Individual adaptation* regarding to each user;
- 2) *Community (or group) adaptation* focusing on a community (or group) of users.

Group adaptation has more advantages than individual adaptation in some situations:

- 1) Common features in a group which are the common information of all members in such group are rela-

tively stable, so it is easy for adaptive systems to perform accurately adaptive tasks.

2) If it is a new user logins system, she/he will be classified into a group and initial information of his model is assigned to common features in such group.

3) In the collaborative learning, users need to learn or discuss together. It is very useful if the collaborative learning is restricted in a group of similar users. Therefore, it is convenient for users that have common characteristics (knowledge, goal, interest, etc.) to learn together because they don't come up against an obstacle when interacting together.

The problem that needs to be solved now is how to determine user groups. This relates to clustering techniques so as to cluster user models because a group is considered as a cluster of similar user models. Sections 2 and 3 discuss about user modeling clustering technique, namely k -means algorithm for vector model, overlay model, Bayesian network. In Section 3, we propose formulations so as to compute the dissimilarity of two overlay models (or two Bayesian network). The k -medoids algorithm and similarity measures such as cosine similarity measure and correlation coefficient are discussed in Section 4. Section 5 is the conclusion.

2. User Model Clustering Method

Suppose user model U_i is represented as vector $U_i = \{u_{i1}, u_{i2}, \dots, u_{ij}, \dots, u_{in}\}$ whose elements are numbers. For instance, if U_i represents user knowledge then U_i is considered as a knowledge vector in which the j^{th} component of this vector is the conditional probability expressing how user master knowledge item j^{th} . Note that U_i is called user model or user vector or user model vector.

Suppose there is a collection of users $\{U_1, U_2, \dots, U_m\}$ and we need to find out k groups so-called k user model clusters. A user model cluster is a set of similar user models, so user models in the same cluster is dissimilar to ones in other clusters. The dissimilarity of two user models is defined as Euclidean distance between them.

$$\text{dissim}(U_1, U_2) = \text{distance}(U_1, U_2) = \sqrt{(u_{11} - u_{21})^2 + (u_{12} - u_{22})^2 + \dots + (u_{1n} - u_{2n})^2}$$

The less $\text{dissim}(U_1, U_2)$ is, the more similar U_1 and U_2 are. Applying k -means algorithm [1], we partition a collection of user models into k user model clusters. The k -means algorithm (see Figure 1) includes three following steps:

1) It randomly selects k user models, each of which initially represents a cluster mean. Of course, we have k cluster means. Each mean is considered as the "representative" of one cluster. There are k clusters.

2) For each remaining user model, the dissimilarities between it and k cluster means are computed. Such user model belongs to the cluster which it is most similar to; it means that if user model U_i belong to cluster C_j , the dissimilarity measure $\text{dissim}(U_i, C_j)$ is minimal.

3) After that, the means of all clusters are re-computed. If stopping condition is met then algorithm is terminated, otherwise returning step 2.

This process is repeated until the stopping condition is met. For example, the stopping condition is that the square-error criterion is less than a pre-defined threshold. The square-error criterion is defined as below:

$$\text{Err} = \sum_{i=1}^k \sum_{U \in C_i} \text{dissim}(U - m_i)$$

where C_i and m_i is cluster i and its mean, respectively; $\text{dissim}(U, m_i)$ is the dissimilarity between user model U and the mean of cluster C_i .

The mean m_i of a cluster C_i is the center of such cluster, which is the vector whose t^{th} component is the average of t^{th} components of all user model vectors in cluster C_i .

$$m_i = \left(\frac{1}{n_i} \sum_{j=1}^{n_i} u_{j1}, \frac{1}{n_i} \sum_{j=1}^{n_i} u_{j2}, \dots, \frac{1}{n_i} \sum_{j=1}^{n_i} u_{jn} \right)$$

where n_i is the number of user model in cluster C_i .

3. Clustering Overlay Model

If user is modeled in a vector, the dissimilarity measure in k -mean algorithm is Euclidean distance. However,

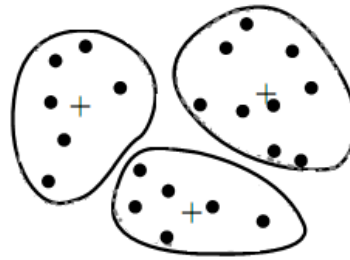


Figure 1. User model clusters (means are marked by sign “+”).

there is a question: how to compute such measure in case that user model is an overlay model which is in form of domain graph? In this situation, the domain is decomposed into a set of elements and the overlay model is simply a set of masteries over those elements. So overlay model is the subset of domain model (see **Figure 2**).

It is essential that overlay model is the graph of domain; so overlay model is also called as graph model. Each node (or vertex) in graph is a knowledge item represented by a number indicating how user masters such knowledge item. Each edge (or arc) reveals the relationship between two nodes. It is clear to say that the dissimilarity measure needs changing so as to compare two overlay models. Note that the terms “user model”, “overlay model”, “graph model” are the same in this context. Suppose two user overlay models U_1 and U_2 are denoted as below:

$$U_1 = G_1 = \langle V_1, E_1 \rangle \text{ and } U_2 = G_2 = \langle V_2, E_2 \rangle$$

where V_i and E_i are set of nodes and set of arcs, respectively. V_i is also considered as a vector whose elements are numbers representing user’s masteries of knowledge items.

$$V_i = (v_{i1}, v_{i2}, \dots, v_{in})$$

Suppose graph model is in form of tree in which each directed arc represents the prerequisite relationship of two nodes. If there is an arc from node A to node B, user must master over A before learning B.

Let $\text{depth}(v_{ij})$ is the depth level of node j of graph model G_i . Note that the depth level of root node is 1.

$$\text{depth}(v_{root}) = 1$$

Given the assumption “*the structure of graphs of all users is kept intact*”, the dissimilarity (or distance) of two graph models G_1 and G_2 is defined as below:

$$\text{dissim}(G_1, G_2) = \text{distance}(G_1, G_2) = \sum_{j=1}^n \left| \frac{v_{1j} - v_{2j}}{\text{depth}(v_{1j})} \right|$$

The meaning of this formulation is: the high level concept (node) is the aggregation of low level (basic) concepts.

The depth levels of j^{th} nodes of all graph models are the same because the structure of graphs is kept intact.

$$\forall a, b, j, \text{ depth}(v_{aj}) = \text{depth}(v_{bj})$$

For example, there are three graph models whose structures are the same to the structure shown in **Figure 3**. The values of their nodes are shown in **Table 1**:

The dissimilarity (or distance) between G_3 and G_1, G_2 , respectively are computed as below:

$$\text{dissim}(G_1, G_3) = \left| \frac{2-2}{1} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{0-1}{3} \right| + \left| \frac{3-4}{3} \right| + \left| \frac{2-5}{3} \right| + \left| \frac{1-4}{3} \right| = 2.66$$

$$\text{dissim}(G_2, G_3) = \left| \frac{1-2}{1} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{0-1}{2} \right| + \left| \frac{1-1}{3} \right| + \left| \frac{4-4}{3} \right| + \left| \frac{5-5}{3} \right| + \left| \frac{4-4}{3} \right| = 1.5$$

So G_2 is more similar to G_3 than G_1 is.

3.1. In Case That Arcs in Graph Are Weighted

In case that each arc is assigned a weight representing the strength of prerequisite relationship between parent

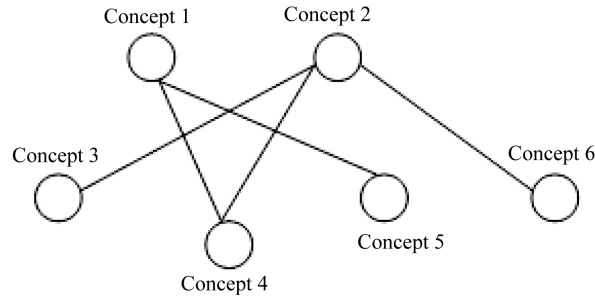


Figure 2. Overlay model.

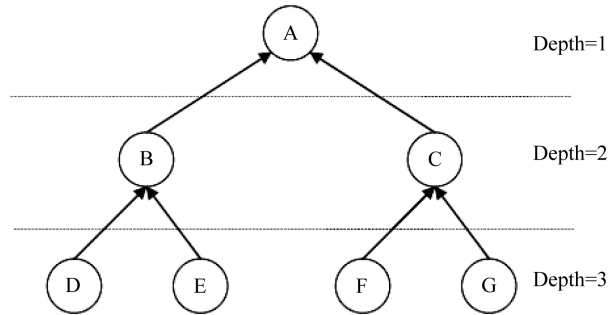


Figure 3. Graph model in form of tree and prerequisite relationships.

Table 1. Values of graph nodes.

	A	B	C	D	E	F	G
G_1	2	1	1	0	3	2	1
G_2	1	1	0	1	4	5	4
G_3	2	1	1	1	4	5	4

node and child node, the following figure shows this situation:

The dissimilarity (or distance) of two graph models G_1 and G_2 is re-defined as below:

$$\text{dissim}(G_1, G_2) = \text{distance}(G_1, G_2) = \sum_{j=1}^n \left| \frac{v_{1j} - v_{2j}}{\text{depth}(v_{1j})} \times \text{weight}(v_{1j}) \right|$$

Let $\text{weight}(v_{ij})$ be the weight of arc from node j (of graph model i) to its parent. We consider that $\text{weight}(v_{ij})$ is the weight at node v_{ij} . The sum of weights at nodes having the same parent is equal to 1.

$$\sum_{v_{ij} \text{ have the same parent}} \text{weight}(v_{ij}) = 1$$

The weight at root node is equal to 1.

$$\text{weight}(v_{root}) = 1.$$

The weight at j^{th} nodes of all graph models are the same because the structure of graphs is kept intact.

$$\forall a, b, j, \text{weight}(v_{aj}) = \text{weight}(v_{bj})$$

For example, there are three graph models whose structures are the same to the structure shown in Figure 4. The values of their nodes are shown in Table 1. The dissimilarity (or distance) between G_3 and G_1 , G_2 , respectively are computed as below:

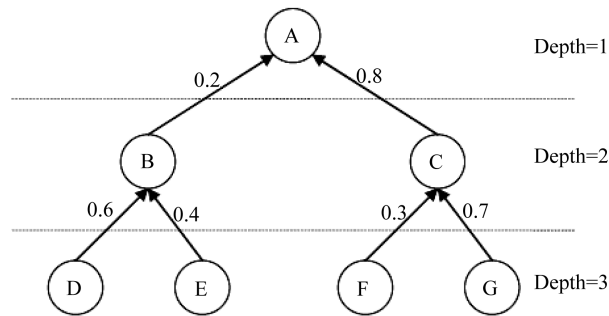


Figure 4. Graph model and weighted arcs.

$$\begin{aligned} \text{dissim}(G_1, G_3) &= \left| \frac{2-2}{1} \right| + \left| \frac{1-1}{2} \times 0.2 \right| + \left| \frac{1-1}{2} \times 0.8 \right| + \left| \frac{0-1}{3} \times 0.6 \right| \\ &\quad + \left| \frac{3-4}{3} \times 0.4 \right| + \left| \frac{2-5}{3} \times 0.3 \right| + \left| \frac{1-4}{3} \times 0.7 \right| = 1.33 \\ \text{dissim}(G_2, G_3) &= \left| \frac{1-2}{1} \right| + \left| \frac{1-1}{2} \times 0.2 \right| + \left| \frac{0-1}{2} \times 0.8 \right| + \left| \frac{1-1}{3} \times 0.6 \right| \\ &\quad + \left| \frac{4-4}{3} \times 0.4 \right| + \left| \frac{5-5}{3} \times 0.3 \right| + \left| \frac{4-4}{3} \times 0.7 \right| = 1.4 \end{aligned}$$

So G_1 is more similar to G_3 than G_2 is.

3.2. In Case That Graph Model Is Bayesian Network

Bayesian network is the directed acyclic graph (DAG) [2] [3] constituted of a set of nodes and a set of directed arc. Each node is referred as a binary variable and each arc expresses the dependence relationship (namely, prerequisite relationship) between two nodes. The strength of dependence is quantified by Conditional Probability Table (CPT). In other words, each node owns a CPT expressing its local conditional probability distribution. Each entry in the CPT is the conditional probability of a node given its parent. The marginal probability of a node expresses user's mastery of such node. Note that marginal probability is considered as posterior probability.

Suppose the structure of Bayesian network is in form of tree, Figure 5 shows our considered Bayesian network.

Let G_1 and G_2 be two Bayesian networks of user₁ and user₂, respectively.

$$G_1 = \langle V_1, E_1 \rangle \text{ and } G_2 = \langle V_2, E_2 \rangle$$

where V_i and E_i are a set of nodes and a set of arcs, respectively. The dissimilarity (or distance) of two Bayesian networks G_1 and G_2 is defined as below:

$$\text{dissim}(G_1, G_2) = \text{distance}(G_1, G_2) = \sum_{j=1}^n \left| \frac{Pr(v_{1j}) - Pr(v_{2j})}{\text{depth}(v_{1j})} \right|$$

where $Pr(v_{ij})$ is the marginal probability of node j in network i . The inference technique to compute marginal probability is discussed in [2].

4. Similarity Measures for Clustering Algorithm

Although dissimilarity measure considered as the physical distance between them is used to cluster user models, we can use another measure so-called similarity measure so as to cluster user models. There are two typical similarity measures:

- 1) Cosine similarity measure
- 2) Correlation coefficient

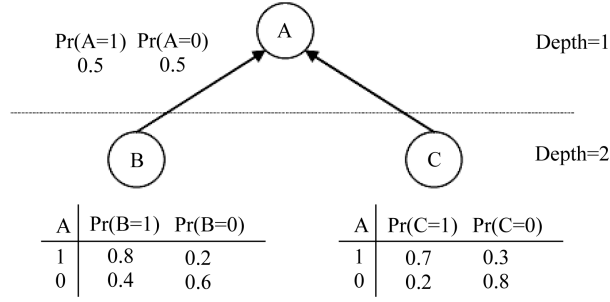


Figure 5. Bayesian network and its CPT (s).

Suppose user model U_i is represented as vector $U_i = \{u_{i1}, u_{i2}, \dots, u_{ij}, \dots, u_{in}\}$, the cosine similarity measure of two user models is the cosine of the angle between two vectors.

$$\text{sim}(U_i, U_j) = \cos(U_i, U_j) = \frac{U_i \cdot U_j}{|U_i| \cdot |U_j|} = \frac{\sum_{k=1}^n u_{ik} \times u_{jk}}{\sqrt{\sum_{k=1}^n u_{ik}^2} \sqrt{\sum_{k=1}^n u_{jk}^2}}$$

The range of cosine similarity measure is from 0 to 1. If it is equal to 0, two users are totally different. If it is equal to 1, two users are identical. For example, **Table 2** shows four user models.

The cosine similarity measures of user_4 and user_1 , user_2 , user_3 are computed as below:

$$\text{sim}(\text{user}_4, \text{user}_1) = \frac{1 \times 1 + 2 \times 2 + 1 \times 0}{\sqrt{1^2 + 2^2 + 0^2} \sqrt{1^2 + 2^2 + 1^2}} = 0.9$$

$$\text{sim}(\text{user}_4, \text{user}_2) = \frac{2 \times 1 + 1 \times 2 + 1 \times 0}{\sqrt{1^2 + 2^2 + 0^2} \sqrt{2^2 + 1^2 + 2^2}} = 0.6$$

$$\text{sim}(\text{user}_4, \text{user}_3) = \frac{4 \times 1 + 1 \times 2 + 5 \times 0}{\sqrt{1^2 + 2^2 + 0^2} \sqrt{4^2 + 1^2 + 5^2}} = 0.4$$

Obviously, user_1 and user_2 are more similar to user_4 than user_3 is.

On other hand, correlation coefficient which is the concept in statistics is also used to specify the similarity of two vectors. Let \bar{U}_i be the expectation of user model vector U_i .

$$\bar{U}_i = \frac{1}{n} \sum_{k=1}^n u_{ik}$$

The correlation coefficient is defined as below:

$$\text{sim}(U_i, U_j) = \text{correl}(U_i, U_j) = \frac{\sum_{k=1}^n (u_{ik} - \bar{U}_i)(u_{jk} - \bar{U}_j)}{\sqrt{\sum_{k=1}^n (u_{ik} - \bar{U}_i)^2} \sqrt{\sum_{j=1}^n (u_{jk} - \bar{U}_j)^2}}$$

The range of correlation coefficient is from -1 to 1 . If it is equal to -1 , two users are totally different. If it is equal to 1 , two users are identical. For example, the correlation coefficients of user_4 and user_1 , user_2 , user_3 are computed as below:

$$\bar{U}_4 = \frac{1+2+0}{3} = 1, \quad \bar{U}_1 = \frac{1+2+1}{3} = 1.33, \quad \bar{U}_2 = \frac{2+1+2}{3} = 1.66, \quad \bar{U}_3 = \frac{4+1+5}{3} = 3.33$$

$$\text{sim}(\text{user}_4, \text{user}_1) = \frac{(1-1)(1-1.33) + (2-1)(2-1.33) + (0-1)(1-1.33)}{\sqrt{(1-1)^2 + (2-1)^2 + (0-1)^2} \sqrt{(1-1.33)^2 + (2-1.33)^2 + (1-1.33)^2}} = 0.86$$

Table 2. Four user models.

	feature ₁	feature ₂	feature ₃
user ₁	1	2	1
user ₂	2	1	2
user ₃	4	1	5
user ₄	1	2	0

$$\begin{aligned}
& \text{sim}(\text{user}_4, \text{user}_2) \\
&= \frac{(1-1)(2-1.66) + (2-1)(1-1.66) + (0-1)(2-1.66)}{\sqrt{(1-1)^2 + (2-1)^2 + (0-1)^2} \sqrt{(2-1.66)^2 + (1-1.66)^2 + (2-1.66)^2}} \\
&= -0.86 \\
& \text{sim}(\text{user}_4, \text{user}_3) \\
&= \frac{(1-1)(4-3.33) + (2-1)(1-3.33) + (0-1)(5-3.33)}{\sqrt{(1-1)^2 + (2-1)^2 + (0-1)^2} \sqrt{(4-3.33)^2 + (1-3.33)^2 + (5-3.33)^2}} \\
&= -0.96
\end{aligned}$$

Obviously, user₁ and user₂ are more similar to user₄ than to user₃.

If cosine measure and correlation coefficient are used as the similarity between two user vectors, the serious problem will occur. That is impossible to specify the mean of each cluster because cosine measure and correlation coefficient have different semantic from Euclidean distance. Instead of using k -mean algorithm, we apply k -medoid algorithm [1] into partitioning a collection of user models into k user model clusters. The “mean” is replaced by the “medoid” in k -medoid algorithm. The medoid is the actual user model and its average similarity to all other user models in the same cluster is maximal (see Figure 6).

1) It randomly selects k user models as k medoids. Each medoid is considered as the “representative” of one cluster. There are k clusters.

2) For each remaining user model, the similarities between it and k medoids are computed. Such user model will belong to cluster C_i if the similarity measure of this user model and the medoid of C_i is the highest.

3) After that, k medoids are selected again so that the average similarity of each medoid to other user models in the same cluster is maximal. If k medoids aren’t changed or the absolute-error criterion is less than a pre-defined threshold, the algorithm is terminated; otherwise returning Step 2.

Figure 7 is the flow chart of k -medoid algorithm modified with similarity measure:

The absolute-error criterion which is the typical stopping condition is defined as below:

$$\text{Err} = \sum_{i=1}^k \sum_{U \in C_i} (1 - \text{sim}(U - \text{medoid}_i))$$

where C_i and medoid_i is cluster i and its medoid, respectively; $\text{sim}(U, \text{medoid}_i)$ is the similarity of user model U and the medoid of cluster C_i .

5. Conclusions

We discussed two clustering algorithms: k -means and k -medoids. It is concluded that the dissimilarity measures considered as distance between two user models are appropriate to k -means algorithm and the similarity measure, such as cosine similarity measure and correlation coefficient are fit to k -medoids algorithm. The reason is that the essence of specifying the mean of cluster relates to how to compute the distances among user models. Conversely, the cosine similarity measure and correlation coefficient are more effective than distance measure because they pay attention to the direction of user vector. For example, the range of correlation coefficient is from -1 , “two users are totally different” to 1 , “two users are identical”.

However, cosine similarity measure and correlation coefficient are only used for vector model; they can’t be applied into overlay model, Bayesian network model. It is clear to say that the formulations we proposed to

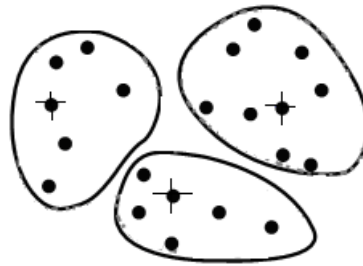


Figure 6. k -medoid algorithm (user model vectors considered as medoids are marked by sign “+”).

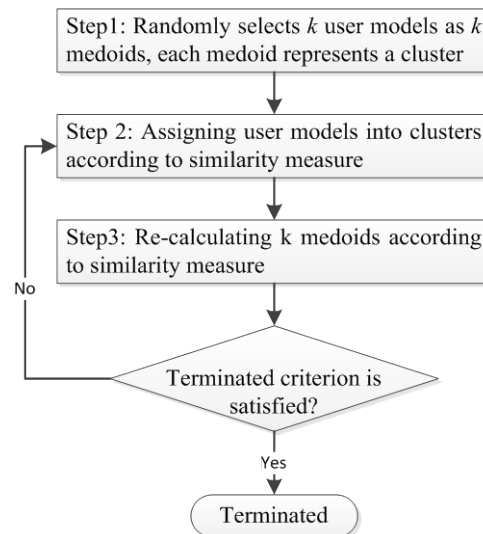


Figure 7. Flow chart of k -medoid algorithm.

compute the dissimilarity of two overlay models (or Bayesian network) are the variants of distances between user models.

References

- [1] Han, J.W. and Kamber, M. (2006) Data Mining: Concepts and Techniques. Second Edition, Elsevier Inc., Amsterdam.
- [2] Nguyen, L. and Phung, D. (2009) Combination of Bayesian Network and Overlay Model in User Modeling. *Proceedings of 4th International Conference on Interactive Mobile and Computer Aided Learning (IMCL 2009)*, 22-24 April 2009, Amman.
- [3] Neapolitan, R.E. (2003) Learning Bayesian Networks. Northeastern Illinois University Chicago, Illinois. Prentice Hall, Upper Saddle River.