EXPLOITING THE CONSTRUCTION OF E-LEARNER COMMUNITIES FROM A TRUST CONNECTIONIST POINT OF VIEW

Fan Yang
Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China

Bernd J. Krämer
Faculty of Electrical and Information Engineering, FernUniversität in Hagen, Germany

Peng Han, Ruimin Shen
Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China

E-learning settings such as on-line courses offered in China often involve large numbers of geographically dispersed students who have diverse professional background, learning preferences, and disparate learning needs. However, current e-learning and classroom teaching methods are extremely limited with respect to personalized learning as they typically provide the same content to all students. Hence, students are finding it difficult to make a decision about which learning materials best meet their personal demands, whilst instructors are finding it almost impossible to recommend personalized materials most appropriate to students since the individual study situations are disparate and hard to analyze. This paper proposes an e-learner Community Building strategy based on Hebbian Learning Rule, which helps learners to set up trusted neighbor connections to peer students with similar study situations. In addition, students can get useful recommendations for proper learning resources from their peers. Based on this theory a recommendation platform has been developed that enables a learner to enquire recommendations, vote on recommended resources and communicate with neighbor learners. Experimental results derived from real learner data have shown that this system can organize learners properly and efficiently, and the cooperative recommendation service indeed improves the students’ learning achievements.

Keywords: personalized e-learning, distributed e-learner community, Hebbian learning

1. Introduction

With the rapid growth of the Internet and mobile communication technologies, e-learning environments enable large numbers of students who were ruled out from higher education for different reasons in the past to have access to abundant learning contents with an unprecedented flexibility and convenience. E-learning settings such as on-line courses offered in China often involve large numbers of geographically dispersed students who have diverse professional background, learning preferences, and disparate learning needs. Current e-learning and classroom teaching methods are extremely limited with respect to personalized learning as they typically provide the same content to all students. The Internet, with its plethora of learning resources, seems to provide a better basis for personalized...
learning, but students are finding it difficult to make a decision about which learning materials best meet their demands. Instructors, as well, are finding it almost impossible to recommend personalized materials most appropriate to students because the individual study situations are disparate and hard to analyze. Thus, finding an answer to the question how to provide personalized learning content is of high priority for e-learning applications.

A recent literature review shows that several researchers have attempted to adopt recommender systems to e-learning applications. Shen and Shen (2004) have described a mechanism focused on how to organize the learning materials based on a domain ontology, which can guide the learning resources recommendation according to their learning status. Lu (2004) has proposed a multi-attribute evaluation method to measure a student’s need and he developed a fuzzy matching method to find suitable learning materials to best meet each student need. Lee and Su (2004) have studied a method to model distributed and sharable learning resources based on SCORM (Sharable Content Object Reference Model) (SCORM, 2005), which allows the implementation of personalized content navigation services. Wang and Shao (2004) have developed a personalized recommendation method integrating user clustering and association-mining techniques, which is applied to an e-learning site for learning navigation. (Luo et.al., 2002) presented a method to organize components and courseware using the hierarchy and association rules of the concepts, which can recommend the relative contents to students and also can help them to control the learning schedule.

These recommendation approaches applied to e-learning environments focused on learning objects sequence organization and learning status analysis. However most of these methods are missing one important issue an e-learning Recommender System should address: learning is a social process involving interactivity and communication with peers and tutors concerning proper choices for learning content and activities. We have investigated the behavior of our students in the Network Education College of Shanghai Jiaotong University and found out that learners share common evaluations or needs of learning resources if they have similar learning preferences and situations. From this observation we conjectured that the individual learning effect can be substantially enhanced if an effective method can be presented that helps learners with similar interests to share useful learning resources or learning experiences.

This paper proposes an e-learner Community Exploitation strategy based on Hebbian Learning Rule, which helps learners to set up trusted neighbor connections and get useful recommendations for various learning resource. In our model and prototype implementation, for each real student a Learner Agent is generated to monitor the student’s learning behavior and maintain the connection to trusted learning neighbors. We describe this model and a recommendation platform that enables students to enquire recommendations, vote on recommended resource and communicate with neighbor students. Experimental results from real learner data have shown that this system can organize learners properly and efficiently, and the cooperative recommendation service indeed improves the students’ learning achievements.

The rest of this paper is organized as follows. In Section 2, we describe the architecture of an e-learner community from a connectionist point of view. Section 3 presents the community construction and exploitation strategy we impose on such architecture. In Section 4 we discuss a case study on the use of our learning content recommendation platform. In Section 5 the experimental results of our system are presented and analyzed. Section 6 concludes.

2. Architecture of Trust-based E-Learner Community

In e-learning environments students can help each other to discover new resources by directing classmates, whose preferences and interests they believe to know, to proper findings. Learners with similar preferences or learning status often prefer to collaborate and advise learners in the same community about helpful learning materials, project solutions or learning experiences. Being aware of
this collaboration in e-learning environments, this paper aims to enhance the recommendation accuracy based on the combination of an algorithm exploiting the given community structure and the learners’ participation in the recommendation process. Inspired by social networks, we believe that there must be learning-oriented social connections between learners, which illustrate the relationship between learners with similar learning preferences and status.

In our former work, we have developed a system that exploits the learning relationships between learners based on their dynamic request for learning content and self-organizes learners with similar interests into various learning community (Yang et.al., 2003). In this approach we generated a learner agent (LA) acting on behalf of a real student and a group agent (GA) serving as a broker for matching requests from a smaller community of LAs. Each LA monitored and mapped the dynamic learning behavior to one kind of learning resource request. The GA managing a particular LA found the learners with similar resources and moved these learners into one group based on novel exchange and award rules. The experimental results illustrate that this method can cluster the learners quickly and facilitate appropriate organization between learner agents and group agents. But this method only shows a good learner clustering effect when each learner has one primary interest. That is to say, the GA-based community exploitation and management strategy is becoming unstable and ineffective for multi-interest learners.

In order to overlap various interests held by learners, it is better for learners to communicate with each other freely and manage the community relationships by themselves. Inspired by the architecture of peer-to-peer (P2P) systems, we now consider each learner as a peer who is in charge of neighbor list maintenance and trusted weight updating. Each peer can access our e-learning platform via any cyber-device including PC, laptop, PDA, cell phone or others. Peers can communicate directly with each other and need not connect via a server. Furthermore the P2P architecture and routing algorithm enable peers to find a specific resource within a certain time period.

“Trust-based E-Learner Community” is defined as a group of learners who share common learning status and mutually recommend each other related learning resources that meet their learning needs. Each learner has different trust strength on the neighbors. Figure 1 depicts our trust-based e-learner community architecture.

![Fig. 1 Trust-based e-learner community architecture.](image)

For each learner peer $i$, we generate a Learner Agent (LA) $\alpha_i$ to monitor learning behavior, generate a learner profile and maintain relationships with trusted neighbors. The local user data

*Transactions of the SDPS*
maintained by \( a_i \) includes the local Information Resource Set, Resource Vote Vector and Neighbor List. The **Information Resource Set (IRS)** is a memory of observed resources, and collected but unread resources. The **Resource Vote Vector (RVV)** \( V^i = (v^i_1, ..., v^i_d, ..., v^i_L) \) includes a vote value \( v^i_d \) of learner \( i \) on resource \( d \) for \( d \) ranging from 1 to \( L \). The values \( w_{ij} \), which represent the trust strengths from learner \( i \) to learner \( j \) and which are taken from the interval \([0, 1]\), occur as labels on the edges connecting \( i \) to its neighbors \( j \). The **Neighbor list (NL)** maintains the trusted neighbors’ identification and trusted weight \( w_{ij} \), which can be divided into two sets: the Trusted Neighbors Set \( T^i \) and the Potential Neighbors Set \( P^i \); the maximal capacities of \( T^i \) and \( P^i \) are \( MaxT \) and \( MaxP \), respectively.

### Summary of Nomenclature

- \( a \) = learner agent
- \( v^i_d \) = vote on resource \( d \) of learner \( i \)
- \( w_{ij} \) = trusted weight from peer \( i \) to peer \( j \)
- \( T^i \) = trusted neighbor set
- \( P^i \) = potential neighbor set
- \( r^i_j \) = general evaluation on resource \( d \)
- \( ttl \) = time-to-live
- \( PA_{ij} \) = enquire Path from \( a_i \) to \( a_j \)
- \( \Delta w_{ij} \) = increment of trusted weight
- \( \eta \) = learning rate of trusted weight
- \( \beta \) = constant learning speed
- \( k \) = update direction of learning speed
- \( \gamma \) = minimum trusted weight
- \( \mu_r \) = recursive updating rate

### 3. Community Construction and Exploitation Strategy

This paper proposes a novel approach to construct and exploit trust-based e-learner communities relying on the following strategies and learning rules. The Initial Bootstrapping strategy initializes the neighbor list of each peer student and the architecture of the communities. The Recommendation Enquiry and Recommendation Evaluation strategies address the workflow of this system, whilst the means of information forwarding and peer communication are defined by the Spreading Enquiry protocol. Moreover, the Trusted Weight Learning, Trusted Weight Recursive Updating and Neighbor List Updating rules are introduced to allow the system to learn trusted weights between neighbors and correspondingly adjust the community construction.

#### 3.1. Initial Bootstrapping

Peer agents are bootstrapped with neighbors by a peer-to-peer mechanism similar to ICQ\(^*\) with a buddy list and the possibility to send instant messages. The idea is that learners usually have a social network of friends and co-learners whom they can add to their buddy lists with an initial trusted weight. The buddies can then be used as the initial agent’s neighbors set \( T^i \). If the learner never adds any friend to the buddy list during the bootstrapping process, the system will randomly choose several

\(^*\) ICQ is a trademark of ICQ Inc. ([www.icq.com](http://www.icq.com)). ICQ is an instant-messaging program that lets the users set up a buddy list with friends that can be contacted directly with an instant message.

*Journal of Integrated Design and Process Science*
students with similar background as initial neighbors and give each connection an initial trusted weight. In the initial step, the potential neighbor list \( P^i \) is empty.

### 3.2. Recommendation Enquiry

During the learning process, learner \( i \) may access a learning resource and want to know if this resource is worth reading. Thus the learner agent \( a_i \) will deliver a “Recommendation Enquiry” to the learners in \( T^i \) and \( P^i \) according to the Spreading Enquiry protocol described below: “Recommendation Enquiry” could be the title or textual description of the resource and the motivation of this “Recommendation Enquiry” is to get the general evaluation of this resource to help learner \( i \) decide to read it or not.

### 3.3. Recommendation Evaluation

Each neighbor will return the vote on this resource if there is any. Since learner \( i \) has different trust strengths on different neighbors, \( a_i \) will calculate the weight average value based on the recommendation value feedback from the neighbors and give learner \( i \) the general evaluation \( r^i_d \) defined as:

\[
r^i_d = \sum_{j=1}^{m} w_{ij} v^j_d
\]

where \( v^j_d \) is the vote on resource \( d \) by neighbor \( j \) and \( w_{ij} \) is the trust strength from peer \( i \) to \( j \).

Considering this general evaluation \( r^i_d \), learner \( i \) will decide to read the resource or not. Since the background and learning status of the learners are different, the evaluation for a specified resource should vary from learner to learner. Thus the evaluation may be effective only when the learners with similar preferences and status are organized as the trusted neighbors. With the organization of a trust-based e-learner community, the evaluation value should be more similar to the individual assessment of the enquiring learner. In the trusted weight learning process, we ask the each enquiry learner to read the resource and vote on it. Based on the difference between real vote value and feedback evaluation value, we can update the trusted weight between neighbors and exploit the community construction gradually.

### 3.4. Spreading Enquiry protocol

For each LA \( a_i \), the only way to receive an evaluation recommendation is from trusted neighbor agents. Suppose \( a_i \) initiates a recommendation enquiry for resource \( d \); \( a_i \) will first check its neighbor peers to see if any of them has read and voted on resource \( d \). This can be achieved by sending the recommendation enquiry directly to the neighbors in \( T^i \) and \( P^i \), where the Spreading Rule is that a “high weight indicates high probability to be chosen”.

If learner \( j \) has read and voted on this resource \( d \), this neighbor’s learner agent \( a_j \) will return the vote value \( v^j_d \) to \( a_i \). Otherwise it will forward the enquiry to its neighbors according to the Spreading Rule. To avoid traffic overload, we define a parameter \( ttl \) (time-to-live) to denote the number of hops left for the recommendation enquiry. Once a neighbor forwards an enquiry, the \( ttl \) of this enquiry should be minus 1. If an evaluation is found within \( ttl \) forward times, the information provider will feedback the vote value to \( a_i \).
The trusted weight learning algorithm is based on the notion of Feedback Recommendation defined as a triple \((a_j, v_d^j, PA_j)\), where \(a_j\) is the feedback provider agent, \(v_d^j\) is the evaluation vote on resource \(d\) of \(a_j\) and \(PA_j\) is the Enquire Path from \(a_i\) to \(a_j\).

### 3.5 Trusted Weight Learning Algorithm

Suppose there is an evaluation value returned to requester \(a_i\) in \(tl\) forward times, then this enquiry is called an Effective Enquiry \(m\). If the provider \(a_j\) is in the neighbor list of \(a_i\), we call this is a directly-activated enquiry, otherwise indirectly-activated enquiry.

For a peer \(j\), the trust strength of peer \(i\) onto peer \(j\) increases as \(j\) always provides similar recommendation evaluations to \(i\). This is corresponds to the main idea in Hebbian learning, which means that the strength of the action of neuron A onto neuron B increases as A repeatedly participates in firing B.

Hebbian learning was proposed by Donald O. Hebb (Hebb., 1949). It describes how the repeated stimulation of specific receptors leads to the formation of neural assemblies that can act as a closed system after stimulation has ceased. This continuous cerebral activity serves as a prolonged duration for neural structural changes to occur during learning. Hebbian learning is therefore a time-dependent mechanism that modifies synaptic efficacy as a function of pre- and post-synaptic activity (Linsker, 1988; Luger, 2002). In connectionism, Hebbian learning and its variations constitute a type of simple but important learning which adjusts a network's weights to reflect its familiarity with inputs in an unsupervised and competitive manner. Procedures similar to Hebbian learning were also used to form knowledge structures on the web (Bollen, 1996). (Wang, et.al., 2005) also proposes a novel approach to form weighted peer-to-peer networks in a self-organizing and decentralized way.

In this paper, we introduce extended Hebbian learning with a recursion rule and potential neighbors to learn the trusted weight between learners and maintain a form of autonomic connectivity between distributed learners with similar preference and status.

This rule is deployed for each effective enquiry after peer \(i\) voted on resource \(d\). The increment of trust strength from peer \(i\) to peer \(j\) is then updated according to the following equation:

\[
\Delta w_{ij} = \eta x_i y_j
\]  
(2)

where \(\eta\) is the learning rate and \(x_i\) and \(y_j\) denote the input and output of peer \(i\) and peer \(j\), respectively.

The values of \(x_i\) and \(y_j\) are defined as:

\[
\begin{cases} 
  x_i = 1, y_j = 1 & \text{if peer } i \text{ and peer } j \text{ are activated} \\
  x_i = 0, y_j = 0 & \text{if peer } i \text{ and peer } j \text{ are not activated}
\end{cases}
\]  
(3)

Different from the original Hebbian learning rule, we extended the learning rate based on the evaluation difference between the peers instead of the input or output value of peers. The learning rate \(\eta\) can then be calculated according to equation (4):

\[
\eta = k \left( \frac{\beta}{1 + h} \right)
\]  
(4)

where \(h = \frac{1}{1 + \left| v_d^j - v_d^i \right| - N} \). As different application systems may have respective rating scheme, for example, the rating may be an integer between 0 and 5 or a decimal fraction with one effective data between 0 and 1, we define \(N = |V_{max} - V_{min}|\) as the maximum span of the voting scheme. \(\beta\) is a constant learning speed. \(k\) denotes the update direction of the learning speed and could be calculated by equation (5), where \(\alpha\) is a pre-defined parameter based on different rating schemes:

\[
\alpha = a_j
\]  
(5)
3.6. Trusted Weight Recursive Updating Strategy

Suppose the spreading route of \( m \) is \( PA_j : a_i \rightarrow a_{i+1} \rightarrow \ldots \rightarrow a_s \rightarrow \ldots \rightarrow a_j \). Although the peers \( a_x (i < x < j) \) involved in this enquiry did not vote on \( d \), they still make contributions to enquiry \( m \) for forwarding this enquiry to their trusted neighbors and finally finding the relative neighbor. For all peers involved in enquiry \( m \), the trusted weight \( w_{x,x+1} \) from peer \( a_x \) to peer \( a_{x+1} \), \( i \leq x \leq j - 1 \) can be updated as follows:

\[
w_{x,x+1} = w_{x,x+1} + \Delta w_{ij}
\]

where \( \Delta w_{ij} \) is the increment of trusted weight from indirectly-activated enquiry of \( a_i \) to \( a_j \), and \( \mu_r \) is a parameter to fine-tune recursive weight updating. That is to say, for every peer-made contribution to enquiry \( m \), the evaluation information works as feedback to reinforce the strengths of the trusted weights involved. The recursive updating may be positive or negative depending on whether the recommendation evaluation is similar with the enquirer peer.

3.7. Neighbor List Updating Rule

After a valid recommendation enquiry, the recommender agent \( a_i \) will check the status of neighbors in the Potential Neighbor and Trusted Neighbor sets and update the neighbor structure according to the rules described as follows:

(1) Calculate the trusted weight \( \Delta w_{ij} \) from \( a_i \) to \( a_j \), which is an evaluation provider of an indirectly-activated enquiry. If \( \Delta w_{ij} > \gamma \) and \( |P^i| < MaxP \), then add peer \( j \) into the potential neighbors set and the initial trusted weight is:

\[
w_{ij}' = \Delta w_{ij}
\]

where \( \gamma \) is a threshold of the minimum trusted weight.

(2) Select a potential neighbor \( p \) with the highest trusted weight in \( P^i \) if there is any;

- If \( |T^i| < MaxT \), then insert \( p \) into \( T^i \) and delete \( p \) from \( P^i \), i.e., \( T^i = T^i \cup \{p\} \) and \( P^i = P^i - \{p\} \);
- If \( |T^i| = MaxT \), then check if there is a trusted neighbor that should be replaced. That is to say, \( a_i \) should check if there is a trusted neighbor \( q \) consistent to \( \gamma \leq w_{iq} < w_{ip} \). If there is any, then delete \( q \) from \( T^i \) and insert \( p \) into \( T^i \), i.e., \( T^i = T^i - \{q\} \cup \{p\} \) and \( P^i = P^i - \{p\} \).

4. Learning Content Recommendation Platform

Based on the self-organized learner community, the Learning Resource Recommendation Platform is developed to enable personalized recommendation as shown in Figure 2.
This platform has three main panels: ‘Learned Resource List’, ‘Recommended Resource List’ and ‘Neighbors List’ in the left column. A click object in each panel can activate the corresponding panel respectively shown as the right column. The ‘Learned Resource List’ visualizes the learning resources from Information Resource Set (IRS) and enables the learner enquiry for evaluation. For example, if the active learner ‘Totoro’ chooses the resource ‘GroupLens: An Open Architecture’, the associated panel ‘Learned Resource Details’ in the right column will be activated and show the details of the content (shown in Fig. 2). If Totoro wants to know if this resource is worth reading, he can easily click the button ‘Recommendation Enquiry’. In return, the LA monitoring Totoro submits this request to its neighbor agents. Thus the neighbor agents will feedback the general evaluation to Totoro based on the approaches discussed in Section 3. The result is depicted in the “Evaluation” panel in Fig. 2. Since this document has a high evaluation value, Totoro begins to read it. If Totoro votes on this document after he read it, the system will launch the Trusted Weight Learning strategy and Neighbor List Updating Rule to update the construction of trusted neighbors gradually. These processes are transparent to Totoro and other neighbors.

Beyond the recommendation enquiry, this system also enables learners to recommend resources to their neighbors. The ‘Recommended Resource List’ illustrates the resources recommended by other neighbor learners. Also it can stimulate the related panel to show the details of the recommended content and enable the learner to vote on it. In particularly, ‘Neighbors List’ gives the learner community member list and the current learner can choose to communicate with one or several of them. Based on the visualization of neighbors, Totoro also can chose a resource in ‘Learned Resource
List’ or ‘Recommended Resource List’ and recommend it to a specific learner by dragging the resource to the learner’s name in ‘Neighbors List’.

Figure 3 shows the protocol of a case study about the communication between Totoro (inviter) and some of his neighbors. Totoro has read material about content based filtering recommended by Amay and he wants to share this with other neighbors. Ergun wants to read this content and asks Totoro to send it. Totoro can select this document in ‘Recommended Resource List’ with a mouse click and drag it on ‘Ergun’ in his ‘Neighbors List’. Ergun can then see the message synchronously and save the referenced document locally. Other neighbors, such as StrongJohn and David, can discuss with each other via instant messages. Based on interaction and discussion among neighbors, learners can evaluate the satisfaction or similarity of a neighbor’s learning interest. All of the actions generated in this platform can be recorded and analyzed, e.g., to maintain and optimize the community structure.

5. Experimental Results Analysis

We chose 500 volunteers who are students of the Network Education College of Shanghai Jiaotong University. We chose the course ‘Graduate English’ as the experimental subject and prepared a sequence of learning documents. We divided these 500 volunteers into two groups A and B, where each group had 250 students. Students of group A learned in a normal way and could not get any recommendation or guide for the learning documents, while the students of group B were all required to use the online recommendation system. During the learning process, the system generated an LA for each student in group B. Each LA monitored the learning behavior of the related learner,
exploited the community construction and maintained neighbor connections. At the same time, the students of group B could enquire recommendations for specified learning documents from their trusted neighbors. They were also encouraged to recommend supplementary resources via the recommendation platform along with related learning documents. Each student also could receive, read and evaluate the materials recommended from other students.

In order to qualify the efficiency of this particular system, we gave a final exam whose motivation was to certify if the usage of this system really helped students to enhance their learning experience. The detailed achievement comparison between these two volunteer groups is depicted in Figure 4. We can see that the score distributed in [B, A+] is higher than group A, while the score distributed in [D, B-] is just the inverse situation.

![Achievement comparison between two volunteer groups A and B.](image)

**Fig. 4** Achievement comparison between two volunteer groups A and B.

We also did an investigation of the system efficiency evaluation among the 250 volunteers in group B. 73% of the students in this group thought that the system was helpful and efficient, 12% of group B students thought that some recommended resources could not meet their need, while 11% thought that the matched neighbors did not really all show similar interests to themselves. Finally, 4% of group B students felt that the system was not supportive to their study process.

6. Conclusions

An innovative P2P e-learner Community Exploitation strategy based on the Hebbian Learning Rule was proposed in this paper. It helps learners to set up trusted neighbor connections and get useful recommendations for various learning resources. The user-friendly interface of our Recommendation Platform, which was developed to empirically evaluate our model, was also described. The system was shown that its mechanisms can not only enhance the learning resource recommendation accuracy, but also greatly reduce the isolation of distributed learners and increase their learning motivation. In the future, we will consider more complex characteristics and behaviors of learners and extent this strategy to mobile learning.

7. References


