12
Quality Rating and Recommendation of Learning Objects

VIVEKANANDAN KUMAR, JOHN NESBIT, PHILIP WINNE, ALLYSON HADWIN, DIANNE JAMIESON-NOEL, AND KATE HAN

12.1 Introduction

The unceasing growth of the Internet has led to new modes of learning in which learners routinely interact on-line with instructors, other students, and digital resources. Much recent research has focused on building infrastructure for these activities, especially to facilitate searching, filtering, and recommending on-line resources known as learning objects. Although newly defined standards for learning object metadata are expected to greatly improve searching and filtering capabilities, learners, instructors, and instructional developers may still be faced with choosing from many pages of object listings returned from a single learning object query. The listed objects tend to vary widely in quality. With current metadata and search methods, those who search for learning objects waste time and effort groping through overwhelming masses of information, often finding only poorly designed and developed instructional materials. Hence, there is a clear need for quality evaluations prior to making a recommendation that can be communicated in a coherent, standardized format to measure the quality of learning objects.

In the last few years, a number of quality rating standards have been developed. As different evaluation instruments are deployed in learning object repositories serving specialized communities of users, what methods can be applied for translating evaluative data across instruments to allow these data to be shared among different repositories? How can the large number of possible explicit and implicit measures of preference and quality be combined to recommend objects to users? To address these questions, we employ a Bayesian belief networks (BBN), a powerful probabilistic knowledge representation and reasoning technique for partial beliefs under uncertainty. Using BBN, we also tackle problems of insufficient and incomplete reviews in learning object repositories, as well as translating and integrating data among different quality evaluation instruments.

In this chapter, we argue that BBNs are a new way of collecting and analyzing the evaluation of learning object quality, and we present real-world BBNs that are constructed to probabilistically model relationships among different roles of reviewers, among various explicit and implicit ratings, and among items of different evaluation measurements. We also present results from a qualitative study and from simulated testing cases to show that the BBN model makes quantitatively reliable inferences about different dimensions of learning object quality.

12.2 Online Learning and Learning Objects

The competitive nature and ever-increasing productivity of our society creates a demand for "up-to-date," "just-in-time," and "just-enough" learning solutions. Unsurprisingly, the sheer volume and high availability of information on the Internet has led to new methods for learning and knowledge construction in education. Before the Internet era, education was much more reliant on school buildings, classrooms, and face-to-face interaction with teachers, books, and libraries. With the advent of the Internet, many of the educational functions provided by these resources and facilities are made available on-line. Students can learn anywhere, anytime by accessing the Internet. Through this ubiquitous medium, education has become more accessible.

Some regard on-line learning as "the use of Internet technologies to deliver a broad array of solutions that enhance knowledge and performance" [10]. Others accept a broader definition that includes learning through intranets and other electronic networks. Over the last decade, Web-based communication media such as asynchronous and audiographic conferencing technologies have led to important new methods for teaching and learning. The Web also allows access to an ever-richer array of multimedia resources that are mined for integration in Web-based courses. It is the challenges in this latter area to which the research reported in this chapter is addressed.

Substantial funding from both public and private educational and industrial organizations has poured into developing on-line learning resources. Examples of initiatives underway include the Curriculum Online project being undertaken for schools in the United Kingdom at a cost of approximately $500 million (www.curriculumonline.gov.uk), and the Australian Learning Federation, a project similar in emphasis with a $30 million budget (www.thelearningfederation.edu.au). To facilitate access to the many thousands of on-line, multimedia resources, we now require sophisticated databases, technical standards, and network infrastructure. These innovations reach beyond the provision of technical facilities needed to distribute resources. Indeed, the capacity of the Internet for allowing information to be easily shared is affecting the way that learning resources are designed and developed.

Digital resources created within one educational institution can now be distributed and reused on a global scale. However, to fully realize the potential for reuse, there is a need to resolve issues of portability across heterogeneous technical platforms and durability across evolving technologies. The technical standards and implementation criteria for learning objects [1] are intended to address these issues.

12.2.1 Learning Objects

Learning objects are digital resources such as images, documents, and simulations that are designed to meet explicit learning goals [5]. According to one Canadian group a learning object is "any digital resource that can be reused to support learning" [14]. Some sources have more narrow definitions that specify rarely met requirements. For example, at Cisco, the network technology giant, a learning object is composed of a learning objective, metadata, content, practice, and assessment [16]. Properties often attributed to learning objects are modularity, reusability, discoverability, customizability, and interoperability [52].

The modular design of learning objects echoes the "object-oriented" trend in modern computer software development. Object-oriented programming focuses on the creation of software objects that can be more easily aggregated into larger programs and reused within many programs. In principle, learning objects can be created that easily integrate into larger, more complex resources such that changes can be made within the learning object that do not require changes to the aggregating body. In theory, being able to unplug a learning object from its assembling unit, revise it, and replace it saves substantial effort compared to revising conventional course material. It is easier to update a small unit of learning material than it is to update an entire cohesive course. As a result, content management becomes easier. This innovation may become especially valuable as knowledge renewal continues to accelerate.

Closely related to modularity is the important attribute of reusability. Once developed, a learning object can potentially be used in multiple contexts for multiple purposes. Many thousands of schools, colleges, and universities offer similar courses, in which the fundamental knowledge to learned is more or less the same. Yet, most educational institutions develop their own course material. Duplicate investment does not make sense, but in the past, sharing course material was difficult, with time and physical distance constraints. Another difficulty is that all courses contain some elements that are local and not transferable to other contexts because courses are often aimed a specific learner groups and they operate within schools that have different academic and administrative structures. Therefore, courses themselves are not suitable candidates for sharing [18]. Reusable learning objects allow cost-effective sharing to occur at a lower level of granularity than the entire course.

Before a learning object can be reused, it must be discovered. Discoverability is enabled by metadata that describes the object in a standardized format. Discoverability is a nontrivial feature in a Web environment, where massive and all-inclusive information is presented. To aid retrieval, learning objects include, or are represented by, metadata composed of standard attribute fields. The LTSC (Learning Technologies Standards Committee) of the IEEE (Institute of Electrical and Electronics Engineers) has published a metadata standard, called LOM (Learning Object Metadata), that is a standard method for describing and cataloging learning objects to enhance learning object discovery. For example, if all learning objects in a repository have properties of *title*, *keywords*, and *language*, then a learning object search could be constructed as "find a learning object such that its *title* contains 'physics', *keywords* contains 'quantum', in the *language* 'English'." The descriptive and relational information in LOM identifies learning objects so that they can be referenced and searched. LOM is designed to describe any learning object, regardless of its size and content.

Ideally, learning objects are also customizable. Because they are modular, individual learning objects can be redesigned to fit the needs of local organizations and learner populations. Every time an object is customized it is subjected to scrutiny and improvement, resulting in a continuing cycle of quality improvement.

Finally, learning objects have the potential to interoperate with other objects, or with a learning management system using established standards. Interoperability standards are being developed by the IMS (Instructional Management System–Global Learning Consortium) organization that will allow learning objects to communicate basic assessment information to a learning management system through a standard interface [20].

12.2.2 Learning Object Repositories

So that they can be distributed and shared for learning, teaching, and course development, learning objects are stored and indexed in databases called learning object repositories or collections. Most repositories do not store the objects, but rather information about objects that are located on Web servers distributed over the Internet. A typical learning object repository consists of a database storing records that conform to the IEEE metadata standard, and Web forms for submitting and searching object metadata. Repository designers often use guidelines published by national or local organizations (e.g., CanCore, 2003 [55]) that show how to implement subsets of the IEEE standard.

Multiple repositories can be connected together to maximize the pool of objects discoverable through a single search operation. The largest of these federated search sites is the U.S. National Sciences Digital Library (NSDL), which offers federated search of several hundred repositories. Another federated search solution being developed centers around eduSource Canada, a Canadian consortium building a national interoperability framework for both academic and industrial contexts. eduSource Canada presents a framework that unites peer-to-peer and Web services models, and formal mechanisms for interoperability at the transactional and semantic levels. The core of the proposed framework is the eduSource communications layer, an open protocol that enables search, gather, and retrieval (and other IMS digital repository interoperability specifications IMS DRI [61]) within a community of eduSource servers, clients, peers, gateways, and other networks.

Some of the many active learning object repositories are the following:

- MERLOT (www.merlot.org), currently indexing about 12,000 objects, is a repository designed for postsecondary education.
- CAREO (careo.netera.ca) currently indexing about 4000 objects, provides multidisciplinary teaching materials for educators in the Canadian province of Alberta.

TABLE 12.1. Educational elements of IEEE LOM metadata

Interactivity type
Learning resource type
Interactivity level
Semantic density
Intended end user role
Context
Typical age range
Difficulty
Typical learning time
Description
Language

- Maricopa Learning Exchange (www.mcli.dist.maricopa.edu/mlx/index.php) currently indexing about 1200 objects, targeting electronic warehouse of ideas, examples, and resources (represented as "packages") that support student learning.
- Edna Online (www.edna.edu.au) hosts a directory about education and training in Australia along with a database of Web resources for teaching and learning.

12.2.3 Pedagogical Metadata

Table 12.1 lists the educational elements of the IEEE metadata standard as described by CanCore [55].

Some of these elements, such as "semantic density," are rarely used and have been dropped by organizations that set usage guidelines for repository developers [55]. Dissatisfaction with the educational category in the LOM standard has led to proposals for extended forms of pedagogical metadata to make more explicit the educational purposes and methods that learning objects are intended to serve [56, 58].

Carey et al [56] developed a suite of "educational rationale" metatags consistent with the principles of learner centered design. Authors used the metatags to communicate the pedagogical intent underlying the design of the learning objects they create. For instance, an author might incorporate an [apply] tag in a learning object that provides students with practice in the application of theory, or a [monitor] tag in an object that prompts students to evaluate their learning strategies.

Mwanza and Engestrom [58] proposed a taxonomy of seven educational metadata elements derived from activity theory. Examples of elements from their taxonomy are objectives ("subjects shared motives for engaging in activities"), and community ("the environment or context in which objects are created and used"). Although it is doubtful that any single theory of learning or instructional design could find sufficiently broad acceptance to form an international standard for pedagogical metadata, it may be possible to establish a more generalized structure that encompasses descriptive terms from a variety of theories and design practices. For example, there could be a standardized "learning goals" element that encompassed both traditional learning objectives [57] and the activity theory metadata for objectives and desired outcomes described by Mwanza and Engestrom. To extend this scheme, separate fixed vocabularies could be developed for different theories and practices.

Rather than collect pedagogical metadata from learning object authors, one can collect it from teachers and learners. Users of the MERLOT repository [7] are able to post “assignments” for an object that describe lesson plans or educational use cases incorporating the object. The assignment is submitted as a form that has some limited pedagogical metadata fields, such as “learning objectives” and “educational level.” However, much of the information entered into MERLOT assignments is relatively unstructured metadata in open text fields. In general, information in Web-based repositories that describes how specific learning objects have been used is both sparse and unstructured. Although structured metadata, like that supported by the IEEE standard, can be used more precisely to formulate search queries, unstructured metadata may be more appropriate when there is no broadly accepted taxonomy, as is the case with pedagogical metadata.

12.3 Evaluation and Recommendation Systems

As learning object repositories continue to grow, and as more repositories are accessed through federated search facilities, users will become increasingly overwhelmed by the number of objects returned by a search query. Although IEEE-style, descriptive metadata are useful in narrowing search, a point is reached where well-specified searches of precisely cataloged objects return many more objects than can be individually assessed by the user. Pedagogical metadata may help to narrow the field, but ultimately even this additional form of cataloging will not protect users from the “hit shock” produced by viewing an avalanche of search results. The problem is that users do not have a precise enough understanding of what they want to formulate specific queries, and object catalogers can not predict many of the detailed requirements of users. To bridge the knowledge gap between the consumer and producer, we need to introduce additional information about the objects that is generated by third parties.

Third parties, such as other consumers and experts, can generate two important types of information. They can provide quality evaluation of an object, perhaps about how well it functions, or its effectiveness for learning. Or they can offer subjective statement or measures of preference regarding the object. When we purchase a car we might read quality evaluations of different cars that state how well the cars are built or how often they need repair. We also might study sales statistics that indicate the popularity of different cars, or ask car owners how well they like their cars. Quality and preference information provided by third parties will also assist learning object users to narrow their search and avoid hit shock.

In this chapter we regard evaluation systems as social computing tools that are used to create and share information about the quality of learning objects. Such systems may be used to evaluate several specific aspects of learning object quality, such as technical functioning, and usability. Recommendation systems are social computing tools that differ from evaluation systems in that they match learning objects to individual users using information about the users and the objects.

All users of learning objects stand to benefit from evaluations and recommendations. Students, especially, are in need of guidance in selecting appropriate resources. Hill and Hannafin [30] observed that often "students lack sufficient meta-cognitive awareness and comprehension monitoring skill to make effective choices on resources." Third-party information in the form of quality evaluation and recommendation can aid teachers and course developers because it efficiently distributes effort required to examine and compare objects, and it allows these users to draw from the expertise of others to select objects.

12.3.1 Evaluating Quality

Learning evaluation systems are fundamentally composed of (1) tools for generating and storing quality ratings, and (2) a search engine that sorts search results according to averaged ratings in best-first order. The evaluation tools in such systems [eLera, MERLOT] enable reviewers to enter comments and ratings in a Web form. The form is a questionnaire-like instrument that asks reviewers to rate and comment on a set of predefined quality dimensions. In the following discussion and throughout this chapter we use the MERLOT [7] and eLera [8] evaluation systems as examples.

In the MERLOT evaluation tool there are three quality dimensions: quality of content, potential effectiveness as a teaching-learning tool, and ease of use [7]. When evaluating a learning object on MERLOT, for each dimension reviewers comment and provide a rating on a five-point scale. In contrast, there are nine quality dimensions in the Learning Object Review Instrument (LORI) provided on the eLera Web site: content quality, learning goal alignment, feedback and adaptation, motivation, presentation design, interaction usability, accessibility, reusability and standards compliance [9]. In LORI, as in MERLOT, reviewers can comment and rate on a five-point scale.

Demand exists for evaluation instruments with different levels of detail and areas of emphasis. Some users and organizations need detailed quality information in areas such as accessibility for learners with disabilities or compliance with specific industry standards. In other settings, quick and easy evaluations are important to encourage participation. The MERLOT instrument is designed to gather quality evaluations from university faculty who have subject matter expertise and teaching experience, but may not have technical or instructional design expertise. LORI is designed to gather evaluations from a panel of users and specialists with complementary areas of experience and expertise.

Both MERLOT and eLera use quality ratings as a default order for search results. In obtaining the overall score for a learning object, the system can calculate a weighted average over the rated dimensions. Ideally the system could allow searchers to specify the weights used in calculating the quality metric used for ordering the search results. To provide features similar to a recommender system, users might also be able to store their preferred weighting schemes in a preferences tool. We are not aware of any existing evaluation systems with these user-customization capabilities.

In addition to differently weighting quality dimensions, an evaluation system might allow its users to differently weight reviewers. Ratings from reviewers who are more trusted, or are more similar to the user, could be assigned a higher weight in the calculation of an object's quality score. We expect that the measurement and application of *trust metrics* will come to play an increasing role in learning object evaluation, especially in the design of recommendation systems.

At present, there are several practical problems that impede the broad use of learning object evaluation systems. The greatest of these is a supply problem. Despite the benefits offered by high-quality reviews, there are no well-established incentive structures that have been successful in encouraging broad participation in review activities. There is a widening gap between the many thousands of learning objects that are coming on-line every year, and the number of subject-matter specialists, instructional designers, teachers, and learners who are willing to review them. Secondarily, reviewers are often interested in or qualified to review only a subset of quality dimensions presented by a review instrument. Consequently, published reviews may have ratings on only some of the quality dimensions. Finally, reviews created with different instruments (e.g., MERLOT and eLera) are presently incompatible. Even though they may offer comparable information, they cannot be automically combined to generate a single quality score for use in ordering a search. This failing will become increasingly apparent as users turn to federated search facilities that return results from multiple repositories that apply different evaluation criteria.

12.3.2 Recommendation and Trust

Given a set of users and items, where items could be documents, products, other users, etc., a system recommends items to a user based on past behaviour of this user and/or other users, and additional information on users/items [23].

Recommendation is a very common social activity with a broadly understood meaning. However, in social computing, an essential feature of good recommendations is that they are relative to the preferences and needs of the recommendee. The purpose of recommendation systems is to find items that match a specific person or requirement.

Essentially, a recommendation system tries to make the best prediction that a user will or will not like a certain item.

With learning object metadata, better filtering techniques have been developed and applied in learning object discovery, that is, collaborative filtering versus text-based indexing used in most of the Internet search engines. Collaborative filtering holds promise in education not only for the purposes of helping learners and educators find useful resources, but as a means of bringing together people with similar interests and beliefs, and possibly as an aid to the learning process itself [6]

There are two basic types of recommendation systems, content-based and collaborative filtering. There are also social software[1] and social data mining,[2] which are not of our concern here [24].

Content-Based

Content-based recommendation systems use data about the requested item and the information regarding only the active user [25]. Content-based methods, also known as search-based or item-based, treat the recommendation problem as a search for related items. Based on an item provided by a user, the algorithm constructs a search query to find other items with similar keywords or subjects. These items are combined into a recommendation list.

There are different algorithms to determine the most-similar match for a given item. A few popular ones are the vector space model, Bayesian classifiers, and two variants of the vector space model, which are content representation of items and support vector machine classifiers [25].

One of the main limitations with content-based recommendation is that, in a search space, some items' features are impossible to extract for computing predictions. Another limitation is that some important aspects of an item cannot be captured without human intervention. Also, when the search space grows bigger, additional algorithms need to be applied to narrow it down for computation performance purposes [25].

Collaborative Filtering

For collaborative filtering or cluster model, the recommendation is achieved using information about other users, rather than only the active user [26]. In fact, "a pure collaborative recommendation system is one which does no analysis of the items at all—all that is known about an item is a unique identifier" [27]. The main idea is to find a subset of users who have similar tastes to the given user, and use this subset for making predictions. In pure collaborative filtering, one does not need to know the content of an item, only the relationship between different users [25].

There are a number of approaches to implement collaborative filtering, including neighborhood-based, BBN, induction rule learning, and so on. The one most frequently used is the neighborhood-based approach[3] [25].

Traditional collaborative filtering does little or no off-line computation. Using collaborative filtering to generate recommendations is computationally expensive. It is possible to partially address these scaling issues by reducing the data size, either the number of users in the neighbourhood or the number of catalogues within

---

[1] Social software: the recommendation process is supported but not automated [23].

[2] Social data mining: mine log data of social activity to learn group preferences [23].

[3] Neighborhood-based: The main idea is to find a subset of users who have similar tastes to the given user, and use this subset for making predictions.

a user. Unfortunately, these methods also reduce recommendation quality in one way or the other.

In addition to computational difficulty, collaborative filtering methods also suffer from a few other problems, some of which we encounter in learning object recommendations under different contexts:

- *Cold start*: If there are not enough users, it is difficult to find a high similarity coefficient [28].
- *Sparsity*: Sparsity refers to the sparse matrix users/rates, which happens when there are many possible items. This problem is important especially for those similarity functions that take into account only items rated by both users. The sparsity of the matrix could generate only low similarity coefficients or none at all [29].
- *First rater*: It is difficult to give a rate to new items, since they are not rated by anyone [25].
- *Popularity bias*: The system tends to recommend popular items, while being incapable of recommending items to a user with unique taste [25].

People have tried to take the advantages of both content-based and collaborative filtering methods, combing them in a mixed approach. Pazzani [29] used the knowledge of the content for avoiding the problems typical of collaborative filtering, generating more accurate predictions.

Sorting the results list by their quality appears to be an obvious answer, with the highest quality learning object on top of the list. What displays at the beginning of a list naturally gains more attention. This is an intuitive way of communicating with a user for recommendation. Hereafter, for a learning object recommendation we refer to the mechanism to return a list of inquired learning objects, sorted by quality, and the one with the best quality on top of the list.

Another advantage of a learning object recommendation is that the "item," a learning object, is structured, under the supervision of LTSC, but regular recommendation systems have to deal with masses of unstructured, inconsistent information.

From previous reviews of recommendation systems, we can see that generally they have a concept of "trust neighbor." As to what is a "good" neighbor and how to find these neighbors, different recommendation systems apply different algorithms and business implementations. Nevertheless, along with a growing interaction between users and recommendation systems, most systems are able to build up a profile, be it for items in its search space or for individual users. Profiling improves the accuracy of predictions made by recommendation systems. However, a current learning object recommendation is operated differently. Learning objects are first filtered according to LOM, and then sorted by quality rating. Hence, it is not a user preference or taste-related recommendation. A user who is interested in a learning object in discipline X today is quite possibly searching for a learning object in discipline Y tomorrow. There is no preference as to what type of learning objects a particular user is in favor of. It is a completely unpredictable factor in learning object searches. Profiling or the trust-neighborhood approach is not suitable for a current quality-oriented recommendation. In spite of this, it does not stop us from applying collaborative filtering on top of the quality-wise recommendation to make a personalized recommendation.

In this section, we present some of the key features one would employ in the process of implementing a learning object recommendation system.

Quantity

An accurate quality rating on a learning object requires evaluations from different perspectives in a teaching–learning process based on a certain rating standards. It also requires a fair amount of reviews in order to construct a reasonably "true" rating result. More reviews are always desired.

In addition, the resistance of providing reviews also comes from the implementation of the current reviewing system. It is "hard" to use, in a way that if a quality review is not complete, it cannot be submitted. Learning object rating standards are developed by professionals. Very often one reviewer is unable to rate all of the items in a rating standard subjectively. For example, an instructor might not have an accurate rating for motivation, since it is more from a learner's point of view. Similarly, a learner might not have sufficient knowledge to rate standards compliance. This yields two possible results. One is that a user might recoil from submitting a review. The other is that a keen user still submits a review, with randomly rated items that he or she in fact has no opinion on. Therefore, due to a stringent reviewing system, we either lose a potential review, or even worse, we take in a fictional quality rating.

Fairness

Recommendation has always been associated with experts' opinion. This should also apply in a learning object recommendation. For instance, an experienced instructional designer's rating should be taken into more serious consideration than that of a rookie; a subject-matter expert's review should weigh more than that of an anonymous on-line user. Current learning object recommendation systems lack a weighting mechanism whereby evaluations submitted by different reviewers can be taken into account differently, rather than using a simple average value of all quality ratings for a particular learning object. This is especially important when the number of reviews on a learning object is limited. One unfair review could dominate the quality rating and distort the real quality of a learning object.

The other fairness issue is from the recommendation system toward learning objects. In MERLOT, the majority of learning objects are not rated. These unrated learning objects will be returned at the bottom of a result list when they fit the search criteria. They are at the bottom of the list not because of poor quality, but because no review is available. Newly submitted learning objects are not likely to get many reviews, due to short exposure time. If they are returned at the bottom of a recommendation list, they are less likely to be browsed or used. If they are not used, they will not be reviewed. Following this vicious circle, a recommendation system is prejudiced against new learning objects by not giving them a fair start to join the system. This is similar to the first-rater problem in collaborative filtering.

Portability

Much effort has been made to link multiple learning object repositories, for example, the eduSource project [35], whose goal is to build an open network to connect learning object repositories in Canada. This is an inevitable course to ultimately share learning objects over the Internet. With this movement, the recommendation is facing a new challenge.

Learning objects are rated under different measurements: LORI or MERLOT. Ideally, when a search request is sent to a learning object repository, it indexes not only its own storage but also that of connected repositories. The result list is generated, but the recommendation will be impeded by the heterogeneous quality ratings.

There are different ways to solve this problem. Certainly, an industry standard is able to unify quality rating. However, demands exist to have different rating instruments. For instance, LORI and MERLOT serve different levels of detail in rating learning objects. To avoid multiplying evaluation work, with each learning object rated with all available standards, a better solution would be using an invisible adapter that is able to translate one type of rating from the other on the fly. The conversion only needs to be done once, at the "adapter" level. If it requires adjustment in mapping different rating standards or to put in a newly emerged rating instrument, it will not affect any existing evaluation data. Update is also done at the adapter. As such, the rating conversion service for recommendation as well as rating standards mapping maintenance are transparent to learning object repositories, rating standards, and learning object users. This could largely increase the portability of quality rating data.

Integrity

Currently, recommendation relies on quality ratings submitted to a learning object repository. This is a type of explicit rating, where a user's purposeful evaluation is required.

On the other hand, observing how learning objects are used could also provide rich quality-related information, for example, how many times a learning object is requested, how long a user stays with it, how many users put bookmarks on a learning object, how often users come back to use this bookmark, how long it stays as a bookmark, etc. It is similar to using Web page hits to gauge the popularity of a Web site. To some extent, this type of information is more authentic. The preference for certain learning objects is revealed naturally in a relaxed manner. Besides, technologies now exist to make such user activity tracking a reality.

In general, an ideal evaluation system should be flexible and thorough enough to integrate information from varied sources. It should be able to combine data from explicit evaluations as well as implicit measures. Moreover, it is able to mitigate the quantity deficiency in learning object evaluation. With learning objects usage tracking, as long as there are activities on learning objects, whether users review them or not, these activities continuously help learning object repositories to collect quality-related information.

Collecting implicit quality rating data, however, imposes yet another challenge on learning object recommendation systems. That is, how should this implicit information is integrated into the current rating system, so that it could serve the ultimate purpose—quality oriented recommendation?

Based on the above impediment analysis in current recommendation systems, we put forward a few ideas on the general directions to deal with these problems.

Nifty Incentive Mechanism

If users generally do not volunteer to offer quality reviews, learning object repositories should implement some incentive features to encourage or stimulate them to do so, such as greater personal recognition. Although personal recognition sounds vague in a virtual environment like the Internet, there exists a very successful example—Slashdot (www.slashdot.org). It is a Web site where people post, discuss, review, and comment on the latest IT news, issues, and technologies. Its reviewing system applies a concept called "karma," which is a point reward to reviewers who have contributed constructive evaluation to this Web site. People in this Web-site community strive after karma. What can these karma points do? They allow people to review more postings! This is such a benign cycle that one cannot ask for more in a recommendation-dependent system. It is this evaluation system that makes Slashdot a very unique, healthy, dynamic, and informative on-line community.

Hence, to foster an attractive and vigorous electronic culture in the learning object community, this can be a viable solution to encourage more reviews from learning object users.

Effective Implicit Rating Mechanism

The key to evaluation system design is finding ways for explicit ratings and implicit reviews to weave together in a complementary fashion. Compared with what we have achieved in the explicit rating area, we have not done nearly enough in collecting and analyzing explicit reviews in learning object practices. This certainly leaves us with room to grow. To integrate implicit ratings into the current learning object recommendation system, much research and study must be expended in user behavior analysis.

User-Friendly Review System

As we mentioned, learning object rating standards are professionally formulated. This, in a way, intimidates users who are not educational professionals. One way of encouraging evaluation would be to lower the bar of difficulty in using the rating system, for example, installing a system that allows reviewers to rate the learning objects on the items that they are comfortable and confident with. In other words, the system accepts partially rated evaluations.

While review systems present lower barriers to use, they also introduce incomplete rating data into the system. How these values should be integrated into recommendations needs to be resolved.

12.4 Learning Object Quality Rating Using Bayesian Belief Networks

We employ Bayesian belief network [3] (BBN) technology in learning object quality rating and recommendation to address all the concerns listed above.

12.4.1 What We Propose

Quality reviews are quite rare because evaluating the quality of a learning object takes time, effort, and expertise. A reviewer may not be comfortable with evaluating learning objects using every evaluation item listed in a rating standard. Therefore, we designed a recommendation system that accepts partially completed reviews, where reviewers can submit whatever they feel confident about. This increases our confidence in the accuracy of the quality review and hence improves the recommendation performance of the entire system. In the meantime, the system should be able to process this partial review appropriately to augment the accuracy of the quality rating of a learning object.

A good recommendation system should be able to actively collect quality-related information instead of passively waiting for review submission only. Especially in a Web environment, a Web browser's (user's) behavior, which could be tracked by the system, is able to reveal many quality-related issues. These user activities implicitly reflect the quality of learning objects. Our recommendation system is designed to take in these implicit reviews in a proper manner to contribute to the quality rating of a learning object.

Learning objects are evaluated with different rating standards. There are needs for these standards to coexist, serving different disciplines and different user communities. Our recommendation system is designed to convert these different ratings into a unified value so that recommendation can be carried out accordingly. This ability not only facilitates sorting by quality when learning objects are rated with different standards, but also allows a learning object rated with one standard to obtain a quality rating under another. Thus, an on-line learning system with a learning object repository using one rating standard is able to recommend learning objects even if they are returned from an external repository, where a different rating system might be applied.

We also designed the system to weigh quality reviews differently when reviews are submitted from different types of reviewers. For example, a quality review from a group of experts should weigh more than that of a single expert, and an expert's opinion should weigh more than that of an anonymous user.

After studying existing BBN applications and performance in evaluation systems, we find that BBN appears to be a viable solution to equip a learning object quality rating and recommendation system with the features listed earlier. BBN is a powerful probabilistic knowledge representation and reasoning tool for partial beliefs under uncertainty. Uncertainty could be insufficient knowledge, for example lacking of certain quality aspects in a quality review. It combines graph theory and probability theory to provide a practical means for representing and updating probabilities (beliefs) about events of interest, such as the quality rating of a learning object. In addition to offering probabilities of events, the most common task using BBN is to do probabilistic inference, for instance, to infer quality rating from one standard to that of the other.

We present two distinct uses for BBN in our learning object quality rating and recommendation system. First, BBN is used in a single quality review construction to tackle the problem of "incompleteness" of current quality reviews and to unite reviews from using different rating standards. The result is called "unit quality rating." Second, BBN is used to obtain an aggregated rating by integrating reviews from different sources, called "integrated quality rating." This includes different ways of the evaluation system collecting the data, that is, explicit or implicit, as well as the different roles of reviewers who submit the data, that is, recognized experts or anonymous users.

We claim that, through these two BBN approaches, the availability and accuracy of quality ratings can be largely improved in a learning object repository, thus making better learning object recommendations.

The rest of this section presents an introduction to Bayesian belief networks followed by our methodology of quality rating of learning objects in two phases:

In the first phase, we analyze two prevailing learning object quality rating standards: MERLOT[4] and LORI. With consultation from educational experts, we map attributes in one standard to the other, from which we derive a correlation structure between MERLOT Peer Review and LORI. After that, a BBN is constructed from this correlated structure. In this BBN, each rating standard functions as it does individually, giving the closest rating value in one rating standard that has no user review, based on an existing rating of any other attribute rated in either MERLOT or LORI.

For example, we could infer how a learning object would be rated on MERLOT's ease-of-use item, given actual ratings on LORI's interaction-usability and accessibility items. Using probability calculus and Bayes theorem [41], BBN derives the implications of observed events, the rated attributes, by propagating revised probabilities throughout the network, when each attribute's value is updated.

This achieves three things. First, we are able to browse/scan/navigate across learning object repositories to search and recommend learning objects. If a user at the MERLOT Web-site sends out a learning object search request, the MERLOT repository starts its search engine to match the criteria; meanwhile, it sends out a request to a repository where learning objects are rated by LORI. External results come back, whose quality rating data is then entered into the BBN we proposed.

[4] MERLOT: In this section, MERLOT represents MERLOT peer review evaluation criteria.

The outcome from the BBN is the quality rating in MERLOT standard. Thus, these external learning objects can be combined with the MERLOT internal result set. A sorting based on one quality rating can be quickly carried out. The learning objects are returned to the user, and recommended in the order of their quality.

Second, learning object repositories are now able to accept partially rated quality reviews. For example, when using LORI, based on a rating on content quality of a learning object, BBN could produce an "intelligent guess" on what the rating would be for motivation. Although this intelligent guess could only come from a large amount of empirical study, data analysis, and experiment on this BBN, nevertheless the mechanism is set up.

Finally, when constructing the BBN for quality rating using MERLOT and LORI, we decide to give a learning object an initial quality rating value instead of a null value that is neutral, implying neither good nor bad. Any submitted quality review will update this value via BBN propagation. The more quality reviews it gets, the closer this rating value is toward its real quality. A direct result of this is that the newly submitted learning object without quality rating will be returned in the middle of a list rather than behind those that have already been rated poorly. As such, a recommendation system gives a good fair start for new learning objects to join learning object repositories.

In phase one, the quality rating we get out of the BBN is for a single learning object. We call it the unit quality rating. In the second phase, taking a step further, we put a unit quality rating into the bigger picture, regarding where it comes from, how this learning object is used, etc.

As a result, the second BBN is constructed. Through this BBN, a unit quality rating is refined. The weight of an opinion is taken into account depending on who the reviewer is. A panel rating from a group of experts weighs the most. Subsequently, there are individual expert ratings, user panel ratings, individual panel ratings, and anonymous ratings.

Additionally, how a learning object is used also contributes to its quality rating, for instance, how often a learning object is added to a registered user's bookmark list, assuming the on-line learning system has a user interface that allows users to place bookmarks; how long it stays in the bookmark list; how often it is requested; whether it is downloaded or browsed only; and so on. All these activities take a role in the second BBN. They drive a quality rating toward the "real" value from a full view. Thus, we call the result the integrated quality rating. The information from user activity may sound trivial and not very well focused, but the payoff is their high availability and sheer volume. For that reason, we can afford to be strict on what types of activity could contribute to a quality rating and how much they could contribute. This is able to balance out the chaos in user behavior when we use it for a specific purpose.

An integrated quality rating largely depends on how an on-line learning system is implemented and deployed, both on the learning tool itself and its rating mechanism. Therefore, the structure of this BBN is foreseen to be relatively dynamic. It should be built based on the functions and features of a specific on-line learning system.

12.4.2 Bayesian Belief Networks: A Quick Introduction

In this section, we embark on a journey to review Bayesian belief networks (BBNs), a mathematical theory that has been mainly used in knowledge-based planning and scheduling tools in the artificial intelligence domain.

Bayes theorem was developed and named after Thomas Bayes (1702–1761) [41], who first used probability inductively and established a mathematical basis for probability inference. Bayesian belief networks (also known as Bayesian networks, causal probabilistic networks, causal nets, graphical probability networks, probabilistic cause-effect models, and probabilistic influence diagrams) provide decision support for a wide range of problems involving uncertainty and probabilistic reasoning.

The underlying theory of BBN is Bayesian probability theory and the notion of propagation. Although this has been around for a long time, it is only in the last decade that efficient algorithms and tools to implement them have been developed to enable propagation in networks with a reasonable number of variables. The dramatic upswing is visible by looking at the number of books written on Bayesian analysis. During the first 200 years, 1769 to 1969, there were about 15 books written on Bayesian statistics. During 1990 to 1999, roughly 60 Bayesian books have been written, not counting many dozens of Bayesian conference proceedings and collections of papers [36]. The recent explosion of interest in BBN shows that for the first time BBN can be used to solve real-world problems. These recent developments make BBN an excellent method for reasoning about uncertainty. Bayes theorem is expressed as

$$P(H|E,c) = \frac{P(H|c)P(E|H,c)}{P(E|c)}$$

where we can update our belief in hypothesis H given the additional evidence E and the background context c [37]. In the frequentist approach, the probability P of an uncertain event A, $P(A)$, is the frequency of that event based on previous observations. For example, looking at the record of the current champion of badminton, who has a history of eight tournament wins out of 10 times, the probability of a win by the current champion in the next tournament, P (current champion's win) is 0.8. If no such historical observation data exists, Bayesian analysis can reason about beliefs under uncertainty. The expression P(Current Champion's Win | K) thus represents a belief measure, where K implies knowledge about this event (e.g., the player's endurance on the court, techniques of strokes, recent break-up with girlfriend, etc).

One characteristic of Bayes' theorem is $P(H|c)$, which is the probability of the hypothesis H in context c regardless of the evidence. This is referred to as the prior probability. The real power comes when we apply the above theorem to propagate consistently the impact of evidence on the probabilities of uncertain outcomes in a BBN, which will derive all the implications of the beliefs that are input to it. They are usually the facts that can be checked against observations [38].

From a mathematical point of view, a BBN is a directed graph, together with a set of associated probability tables. The graph consists of nodes and arcs. The nodes represent variables, while the arcs represent causal or influential relationships between variables [3].

The BBN is a powerful probabilistic knowledge representation and reasoning tool for partial beliefs under uncertainty. It combines graph theory and probability theory to provide a practical means for representing and updating probabilities (beliefs) about events of interest. The most common task we wish to solve using BBN is probabilistic inference. In addition to the probabilities of events (the probability table), the user knows some evidence, that is, some events that have actually happened, and wishes to infer the probabilities of other events, which have not as yet been observed. Using probability calculus and Bayes theorem, it is then possible to update the values of all the other probabilities in the BBN.

The BBN has an intuitive visual representation, very useful to clarify the opaque problem domain. It not only makes explicit the dependencies between different variables, but also reveals that many of the variables are conditionally independent. However, when the number of variables in the BBN increases, the propagation becomes NP-hard computation. The computational complexity of BBN calculations had severely restricted the number of variables or beliefs that could be modeled, and has prevented the application of BBN to realistic problems [39]. This was the reason that BBN could not be used to solve realistic problems, until later when efficient Bayesian probability algorithm implementation was developed, for example, the Hugin tool [39].

The BBN technology has been applied in various disciplines, including archaeology, economics, education, genetics, law, medicine, quality management, safety management, and risk, management. Well-known application samples include Microsoft Office Assistant Wizard, Microsoft operating system's Technical Trouble Shooter, and SpamBayes (spambayes.sourceforge.net), an anti-email spamming software application.

Human society's development has always been revolving around automating tasks using tools as much as possible. The same applies to recommendation systems. We try to let machines interpret information and do the recommendation for us. Up until now, however, for decision making, in many cases information is still better interpreted by people than by machines. Using BBN in learning object recommendation involves both experts and computers. We try to find a quantitative way to develop qualitative data about information on the Web, thus maximizing both people and computer resources.

12.4.3 Unit Quality Rating

Quality rating, in the scope of our work, is a measure that is used to quantify the quality aspect of a learning object. When we try to provide a tool for quality rating, it is inevitable that we would get involved with the topics in educational measurement and evaluation.

In this research, we do not intend to define learning object quality rating; instead, we simply use it as it has been used in the e-learning community. Hence, in terms of measurement, the conceptualization process is beyond the scope of this project; however, the operationalization process is determined on the participation of two

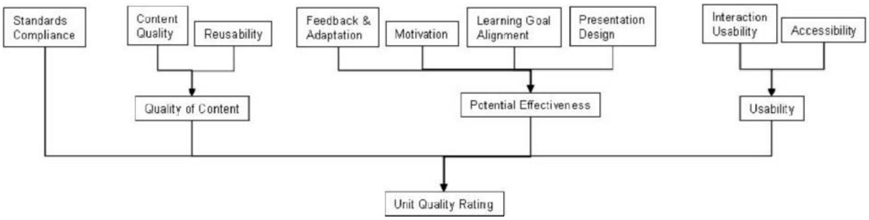

FIGURE 12.1. Unit quality rating structure.

prevailing rating standards, MERLOT and LORI. Both of them apply a scale of number 1 to 5 in measuring the quality of a learning object. As a result, the quality rating from our newly proposed rating mechanism also bears the scale of 1 to 5.

It takes two things to construct a BBN. One is the structure of how all variables or nodes are related—the graph topology. The other is the probability distribution for each variable, known as the node probability table (NPT).

The BBN graph contains a set of nodes and the arcs that link the nodes. As mentioned earlier, one of the difficulties in current learning object recommendation is that different quality rating standards are used to evaluate learning objects. We overcome this problem by integrating MERLOT peer review rating criteria and LORI to obtain a standard neutral quality rating.

In MERLOT, learning object evaluation is based on three dimensions: quality of content, ease of use, and potential effectiveness as a teaching tool. Here we use the notation of, M.QualityOfContent, M.Usability, and M.PotentialEffectiveness.

In LORI, learning object evaluation consists of nine items. Here we prefix with an "L." in front of these items, such as L.ContentQuality, L.Reusability, and so forth.

Figure 12.1 shows the unit quality rating structure that we use to construct the BBN, where MERLOT and LORI are mapped into one. Each item in both rating standards is a node in the BBN. They represent all variables in the learning object quality evaluation domain.

With the graph in Figure 12.1, we then fill the NPT for each node. The NPT contains all the possible values of this node and their distributions. In this case, all nodes, either from MERLOT or from LORI, have 1 to 5 integer values.

In BBN, the nodes that have arcs pointing to them are called child nodes. The source nodes of the arcs are called parent nodes. Each possible value for a node in NPT is contributed by its parents, in the permutation of all their possible values.

If a node has no parent node, for example, at one end of the graph, the NPT we use is a normal distribution.[5] The probability to obtain each value in the set of {1, 2, 3, 4, 5} is {0.05, 0.17, 0.56, 0.17, 0.05}, respectively.

[5] Normal distribution: A normal distribution in a variant χ with mean μ and variance σ^2 is a statistic distribution with probability function

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} E^{-(\phi-\mu)^2/(2\sigma^2)}$$

on the domain χ ™ $(-\infty, \infty)$. It is also called Gaussian distribution or referred to as the "bell curve."

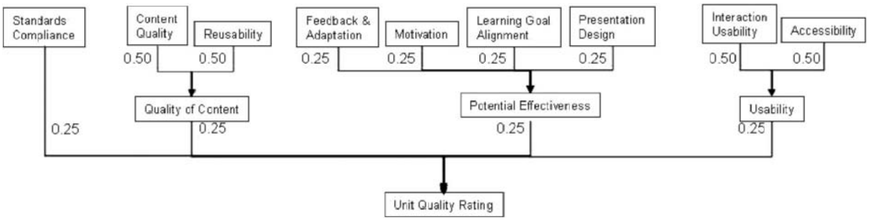

FIGURE 12.2. Unit quality rating BBN.

Thus, we obtain the BBN for unit quality rating, the topology of which is shown in Figure 12.2.

Figure 12.3 is a screenshot of the unit quality rating BBN constructed using JavaBayes [54], along with the NPT for L.ContentQuality node. The Unit_Quality_Rating node is the direct or indirect child of every other node in this BBN, whose value comes from other nodes' propagation of values. Other nodes' values come from quality review either using MERLOT (M) or LORI (L). The value of Unit_Quality_Rating is quality rating standard neutral.

Figure 12.4 is a screenshot of the unit quality rating BBN in JavaBayes with the NPT for the M.ContentOfQuality node. This is where things would get complicated

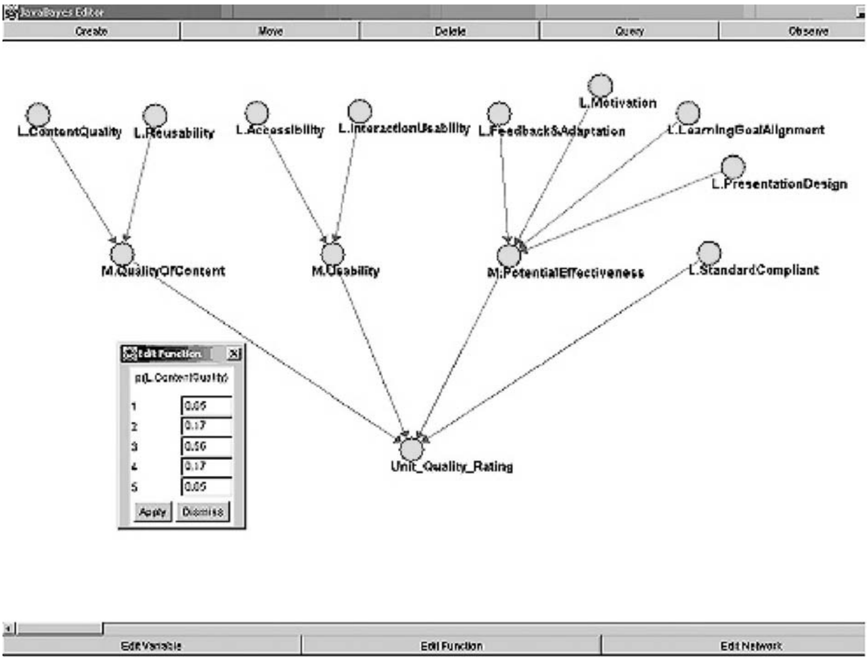

FIGURE 12.3. Unit quality rating BBN with NPT for L.ContentQuality node.

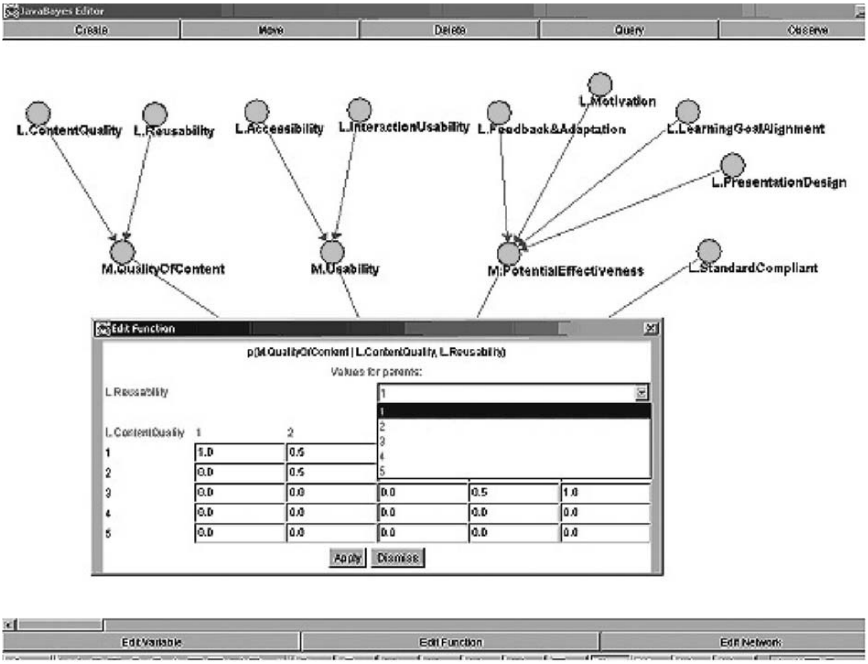

FIGURE 12.4. Unit quality rating BBN with NPT for M.QualityOfContent node.

without a tool like JavaBayes. The node M.QualityOfContent has two parent nodes, L.ContentQuality and L.Reusability.

JavaBayes picks one out of all the parent nodes to display all the possible values, which is 1 to 5. In this case, it is L.ContentQuality. The rest of the parent nodes are accessed via a drop-down list, whose options are all possible values of this parent node. The possible values are all 1 to 5 in this BBN. In Figure 12.4, there is only one drop-down list, L.Reusability. Therefore, the NPT for M.ContentOfQuality enumerates all the probabilities when L.ContentQuality is 1 to 5 under the condition of L.Reusability from 1 to 5. Each of the parent nodes contributes evenly toward the child node's value.

When a node has parent nodes, the NPT depends on whether or not we have empirical knowledge of its parent nodes. If we do, each parent node's influence on this child node can be translated into weight in percentage format. If we do not have any better knowledge about which one of these parent nodes should contribute more than the other, and how much more, as in our presented BBN, then NPT is evenly distributed among the parent nodes. For instance, if the value of L.ContentQuality and L.Reusability are 3 and 4, respectively, the value of M.ContentOfQuality is

$$3 \times 50\% + 4 \times 50\% = 3.5$$

L.ContentQuality and L.Reusability are the only parent nodes M.Content Of Quality has, and there is no prior knowledge of how much L.ContentQuality and

L.Reusability should influence child node M.ContentOfQuality. Therefore, the degree of influence is evenly distributed, which is 50% each.

In the future when we have sufficient quality rating domain knowledge, then parent nodes' influence on a child node can be specified rather than evenly distributed.

The other issue we want to address during constructing this BBN is the "first rater" problem. When a learning object is newly submitted to a repository, it is not likely to get reviews right away. In MERLOT, if this unrated learning object is returned in a search result, it is in the bottom of the list, which is a fundamental flaw.

Assume there are six learning objects, A, B, C, D, E, and Z, returned from one search. Five of them, A, B, C, D, and E, have ratings of 5, 4, 3, 2, and 1, respectively. The sixth one, Z, has no rating.

A user receives a list in the order of {A B C D E Z}. Z appears last due to lack of rating, not because of poor quality, but since it is in the end of the list, it is not likely to be selected by the user. Without being selected by the user, it is not likely to get any review. Without a review, it will remain at the bottom of the list. Thus, the system is biased against new learning objects. It does not get the equal opportunity to be exposed, reviewed, and used.

With BBN, it solves this problem by giving any learning object an initial value as a fair chance to start with. Since normal distribution is applied to the end nodes, the initial quality rating of a learning object will be 3. In a 1-to-5 scale system. Because rating value 3 has 56% out of 100%, all the other values, 1, 2, 4, and 5, have lower probabilities. In above scenario, the user will receive a list in the order of {A B Z C D E} or {A B C Z D E}. Any evaluation that Z gets, BBN then propagates its value and updates the quality rating, which eventually will become more and more accurate to reflect the real quality of this learning object.

12.4.4 *Integrated Quality Rating*

In current reviewing systems, there are expert panel ratings to differentiate reviews from those of regular users, but there is no mechanism to integrate them into one quality rating for a learning object.

In Figure 12.5, the BBN is composed of two types of rating: explicit and implicit. According to the roles of reviewers, we further break down the explicit rating into registered expert rating, registered user rating, and anonymous rating. Under each category some child nodes are listed based on our current understanding.

Note that this type of integrated quality rating largely depends on how a learning system is implemented. For example, some learning object repositories might not employ a group of experts to run a panel rating. It is even more so for implicit ratings. Implicit rating is an important means to get user feedback on learning objects and learning systems, yet how well a system is designed to allow users to expose their preference to learning objects is in the hands of system designers. For instance, certain learning systems' user interfaces allow users to bookmark a learning object. All these links are grouped together, called a bookshelf. When a

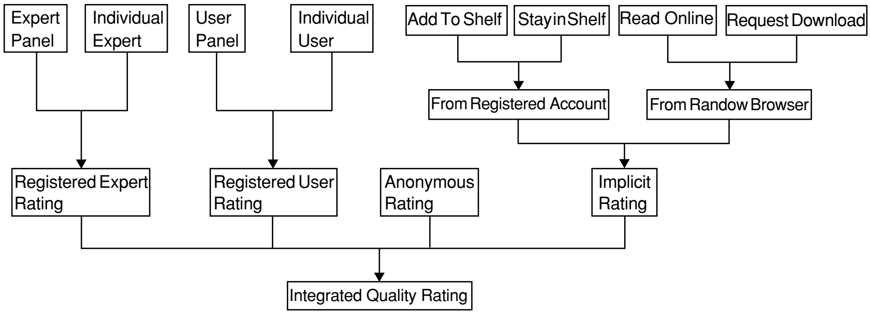

FIGURE 12.5. Integrated quality rating structure.

user comes back to this learning system, he can directly pull out a learning object from the "bookshelf" to use it, much like taking out a book from a real bookshelf. However, some learning systems do not have this mechanism.

Implicit rating not only depends on learning object reviewing and the recommendation system's implementation, but also the system capacity. Implicit rating data usually come from system logging. The more information that is logged, the more substantial analysis one can run on implicit rating, but intensive logging exhausts system resources and takes up database space. How much a system could sustain high-volume logging without impacting system performance varies tremendously. Logging data analysis is another issue. There is no standard logging format. Extracting data related to learning object quality requires a large amount of work in data formatting, sifting, and categorizing.

Above all, user behavior study is fundamental to learning object implicit quality rating. In Figure 12.5, we list items such as AddToShelf and RequestDownload. Certainly there are more user behavior aspects to reflect learning object quality. It requires further study to discover these aspects, as well as to quantify them in relation to learning object quality.

Hence, this part of the chapter describes a way of modeling integrated quality rating of learning objects. We have a long way to go toward obtaining an accurate and precise quality evaluation model.

For the NPT in this BBN, due to the topological structure, it does not make sense to evenly distribute the probability among the parent nodes, such as ImplicitRating and RegisteredExpertRating. For instance, a particular parent node can contribute more than the others. For those nodes that do not have parent nodes, normal distribution is still applied. Thus, by controlling the parent node contribution toward a child node, we ensure the data integrity of the evaluation system. The values in the NPT can be adjusted over time when more data are collected and better knowledge is gained about internode relationships.

Based on the topology in Figure 12.5, an integrated quality rating BBN is constructed, as shown in Figure 12.6.

Figure 12.7 shows split screenshots of the integrated quality rating BBN running in JavaBayes. Note that the darker nodes are with evidence, which means rating

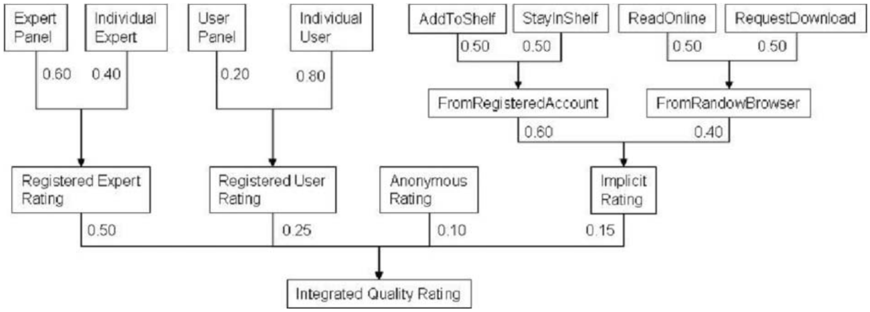

FIGURE 12.6. Integrated quality rating BBN.

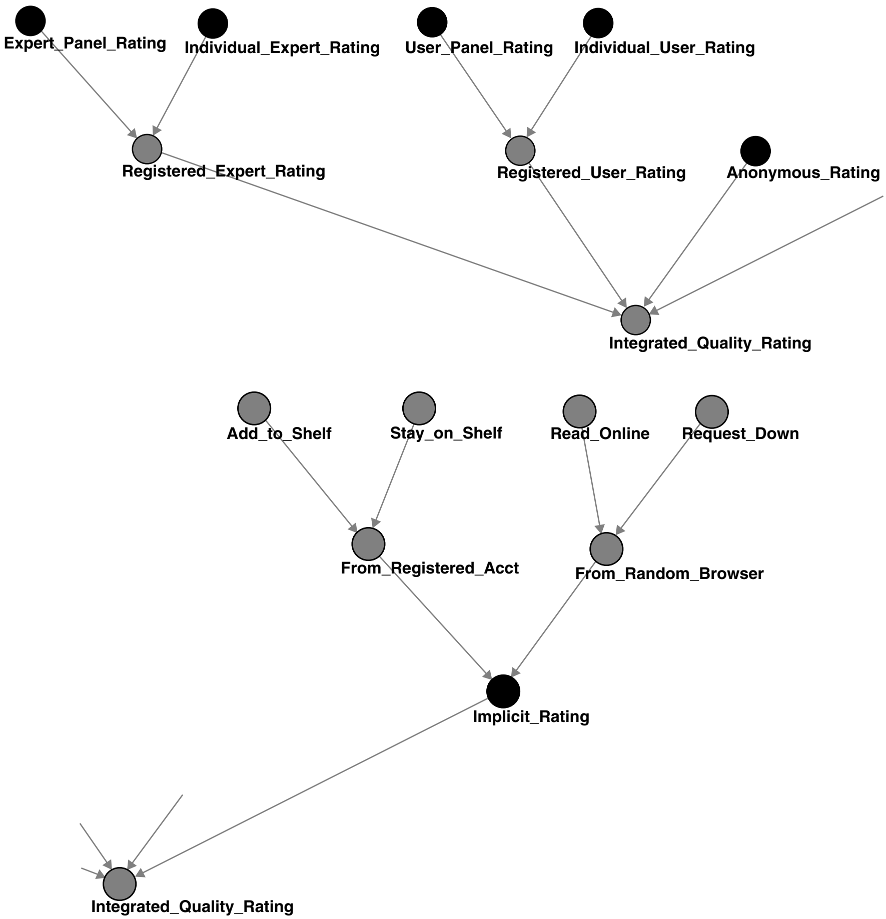

FIGURE 12.7. Integrated quality rating running in JavaBayes.

values are available. The value of other nodes can be inferred from evidence propagation, including integrated quality rating.

12.5 Discussion

We use Bayesian Belief Networks to overcome the incompleteness and absence of learning object quality reviews, as well as the divergence of applied quality rating standards and the monoculture of weighing evaluations from different reviewers. The ultimate goal is to improve learning object quality evaluation and recommendation.

12.5.1 Simulated Test Cases for Individual Rating

In BBN, the process of entering evidence and using it to update the probabilities is called propagation. After the BBN for unit quality rating is created, the inference, mainly the quality rating, can be obtained via the process of propagation.

In JavaBayes, under the "observe" mode, we input the incomplete quality rating we have, that is, the quality rating of three items out of nine items from LORI over a learning object. Then, switch to "query" mode, select Unit_Quality_Rating to observe. The output value, a probability distribution from 1 to 5, is printed out in the JavaBayes console window.

In addition, BBN can be used for both forward and backward inference. This means that if we have a unit quality rating but are missing the details, we can try to infer the quality rating of those items that compose the unit quality rating. What this can bring to us is the comparison and verification of the mapping between different rating standards. For instance, we could have an expert rate a learning object in both LORI and MERLOT. We first enter the LORI evaluation data into BBN, and obtain the detailed rating data for MERLOT rating items via BBN evidence propagation. We then compare MERLOT quality evaluation data from the expert's review and from BBN evidence propagation. If there is any discrepancy and the discrepancy is consistent, we could update the structure in Figure 12.1, or adjust the percentage weight of a parent node toward a child node in BBN in Figure 12.2.

Table 12.2 shows some simulated test cases with incomplete quality reviews as input, and the BBN inference result on various quality rating items.

What we achieved with the unit quality rating BBN was to alleviate the difficulty of quality evaluation by accepting incomplete quality ratings. Users can submit a review only on those quality aspects that they feel comfortable to rate on. Consequently, it leads to a more corresponsive quality rating. With a rating that complies with the actual quality of a learning object, a search result set can be properly sorted and returned in a prioritized manner.

By assigning normal distribution to leaf parent node, an un-rated learning object receives a default unit quality rating value 3, which has highest probability in a 1-to-5 set of probability distribution. Thus, we mitigated the first-rater problem. A learning object that is never rated will not be returned behind poor quality ones with

TABLE 12.2. Results from running unit quality rating BBN

Evidence	Result (inference probability)	Interpolated conclusion
No evidence (a learning object without any rating)	Unit_Quality_Rating: 2.1022165058413246E-4 0.13486437673298943 0.7298508032328528 0.13486437673298948 2.1022165058413246E-4	Unit_Quality_Rating: 3
M.QualityOf Content: 4 M.Usability: 4	Unit_Quality_Rating: 0.0 0.00212041825 0.49787958174999997 0.4978795817500001 0.00212041825	Unit_Quality_Rating: 3.5
M.QualityOf Content: 4 M.Usability: 4	L.InteractionUsability: 0.0 0.021562658548959918 0.3835616438356165 0.40969051243023846 0.18518518518518515	L.InteractionUsability: 4
Unit_Quality _Rating: 4 L.Standard Compliant: 2	M.Usability: 1.7025565641619698E-6 0.004119213762146362 6 0.25868461836582135 0.6223689203228483 0.11482554499261993	M.Usability: 4
Unit_Quality _Rating: 4 L.Standard Compliant: 2	L.Motivation: 0.018327118400728655 0.10008038915613644 0.5248726734433744 0.24916289096414496 0.10755692803561545	L.Motivation: 3

rating lower than 3. It has a fair chance to be exposed, used, and eventually rated. With more evidence, such as quality reviews entering into the BBN, providing most reviewers have conformable evaluations, the initial default bell curve will become narrower and the rating values probability will be less distributed.

Additionally, using BBN's ability to conduct forward and backward inferences, we will be able to correlate the quality rating among different quality attribute nodes from different rating standards, such as MERLOT and LORI. Thus, a learning object query to a MERLOT-rated repository is able to obtain an estimated equivalent quality rating value for the learning objects returned from external learning object repositories that are rated with LORI, and vice versa.

Note that what we get after inference in BBN is a probability distribution among all the possible values, while what we want is one value that has the highest probability. In this project, the rules for interpolation for obtaining unit quality ratings are:

TABLE 12.3. Integrated quality rating BBN testing case

Reviewer Breakdown	Learning Object1	Learning Object2
Expert panel	5	
Expert individual		
User panel		
User		
Anonymous		5
Implicit		
Integrated rating	3.8	3.1

1. The rating that has the highest probability is the rating for the node that is queried.
2. If a few values share the same highest probability, the midpoint value of these values is the rating for the node that is queried.

This is a rough and basic way to interpolate the rating value. More refined and accurate interpolation can be explored and applied in the future.

12.5.2 Simulated Test Cases for Integrated Rating

To test the learning object integrated quality rating BBN, we designed an extreme test case. In this test case, we evaluated two learning objects, Learning Object1 and Learning Object2. For Learning Object1, we have an expert panel rating of 5; the other ratings for Learning Object1 are not available. For Learning Object2, we have an anonymous rating of 5; the other ratings for Learning Object2 are not available. Entering these two values into the BBN, we observed the node of Integrated_Quality_Rating.

Table 12.3 displays the data and result of this testing case.

Since normal distribution is applied to the nodes without parents, the root child Integrated Quality Rating has a value of 3 for a new learning object entering the learning object repository without any rating. In this case, Learning Object1's quality rating is increased from 3 to 3.8 by one expert panel rating of 5, while Learning Object2's quality rating is increased from 3 to 3.1 by one anonymous user rating of 5.

The inference of BBN takes evidence, an observed fact, to update the network with a prior built-in probability distribution. It will not dramatically change a node value with any single evidence entering the system, because we know the overall distribution, and we have control over it. This advantage appears to be more important for the integrated quality rating than for the unit quality rating.

From the above test case, we obtained the probability distributions of the integrated quality rating for these two learning objects. We then plot the resulting probability distributions. Figure 12.8 displays the plotting chart. It shows the positive shifting of the equality rating by these two different types of reviewers, compared to the default equality rating distribution. From a high expert rating, the overall quality rating is increased with an obvious margin, whereas a high anonymous rating has an insignificant impact on the overall rating.

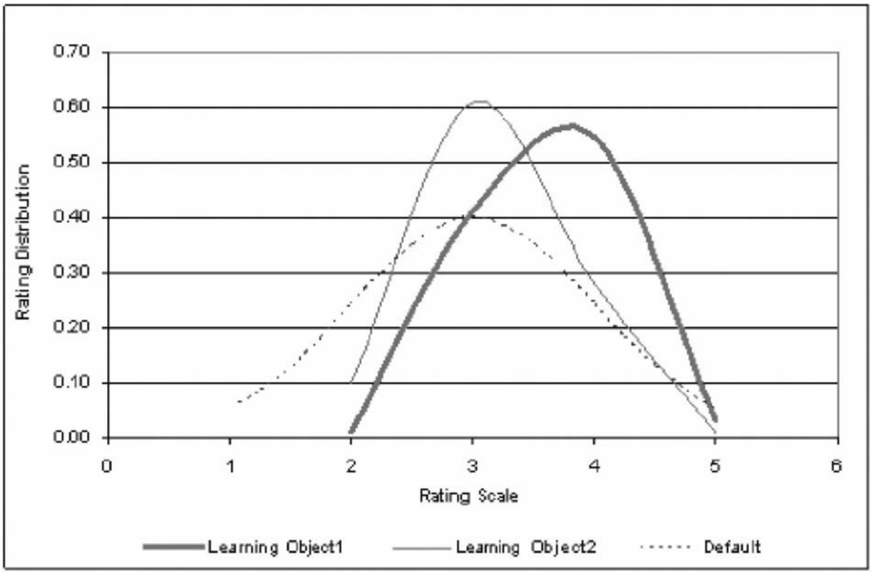

FIGURE 12.8. Analysis of an extreme test case.

This is a desired result when an irresponsible user, usually with anonymous identity, rates a particular learning object, which could be submitted by the user himself, with a high rate over and over again. BBN reduces the negative impact to learning object recommendation from distorted rating to a minimum degree.

This is an essential feature of an evaluation system, but many on-line rating systems do not have this type of behavior. For instance, on the "epinions" Web site (www.epinions.com), assuming two items A and B, under any category, based on a certain standard, A has one review rating of 5, which could be rated by the author or seller himself. B has a couple of ratings of 4, which come from various reviewers. As a result, A tops B when returned as a result of a query. This cannot be considered a healthy rating system.

Nowadays most on-line reviewing systems, as well as learning object repositories, implement a membership mechanism but accept anonymous reviews from unregistered users. It is a desirable feature to accept anonymous reviews for a learning object repository, but it also allows a user to abuse the system by rating a learning object again and again with an untruthful evaluation. However, in applying BBN, this impact can be minimized. With a membership mechanism, each member is allowed to submit one quality review per learning object; we assume registered members are less likely to behave improperly. An anonymous user or author has a higher possibility of rating his favorite item with the highest value, but that does not change the integrated quality rating much. On the contrary, the rating value will be strengthened more by members' evaluation. Hence, even if a learning object repository accepts anonymous reviews in which some are false, the BBN quality rating system still can maintain its integrity by controlling the NPT of anonymous peer ratings to a less influential degree in generating an integrated quality rating.

12.5.3 Reliability and Validity of Our Approach

Reliability is the extent to which we measure some attribute in a systematic and therefore repeatable way [42]. It means that the result would be consistently the same if a test were performed over again. When we use conventional methods to assess the reliability of the conceptual underpinnings of our work we have a promising outcome.

First, in the case of testing without directly involving human examinees, such as converting a quality rating from one standard to the other, the project has perfect test-retest reliability. In the case of testing involving reviewers, for example, accepting partially rated reviews into the system, we think that the reliability is determined by the quality rating standard being used rather than by the quality rating BBN itself.

Second, using internal consistency, the simulated test shows a positive consistency. For example, if "accessibility" has a low rating using LORI standard, the inferred rating of "usability" using MERLOT via the quality rating BBN is also low, and vice versa.

Third, using interrater, a valid testing case would have the same reviewers rate the same learning object using different rating standards, entering the reviews into the quality rating BBN and observing the unit quality rating that comes out, whether they are the same or close. Additionally, due to the empirical characteristic of this project, this type of testing case is supposed to be continuously conducted to help adjust the quality rating BBN.

Lastly, using alternate form, a valid testing case would be obtaining a complete quality review from a reviewer using one rating standard, for example, MERLOT, as well as a partial rating on a particular quality item using another rating standard, for example, LORI. The full quality review in MERLOT is then entered into quality rating BBN, and then an inferred rating on that particular quality item can be obtained in LORI. Comparing the rating from the reviewer and from BBN inference, the closer the values are in alternate form reliability verification, the better. This testing has been conducted as part of the quantitative analysis described earlier.

We would like to point out that, in our work, the interpretation of specific rating values is not important to the recommendation process; rather, the relativity among the learning objects ratings is important. In the test case (above), as long as the two values from different sources are close, it gives a good indication of the reliability of the BBN.

Validity is the extent to which the test being used actually measures the characteristic or dimension we intend to measure [43]. Results are positive when we use conventional methods to measure the validity of this project.

In terms of face validity, educational experts were consulted during quality rating BBN construction.

In terms of content validity, this links back to the quality rating‘s conceptualization and operationalization processes, which are beyond the scope of this chapter.

In terms of criterion validity, we have conducted a very successful simulated test case that is a typical problem in real-world on-line rating applications. The result shows an evident strength to solve this type of problem when using BBN. It will be discussed in later sections of this chapter.

Construct validity is known as the most difficult validity to achieve. According to Hunter and Schmidt [43], construct validity is a quantitative question rather than a qualitative distinction such as "valid" or "invalid"; it is a matter of degree. Therefore, in this work it will take a large amount of tests, experiments, and iterative modifications for a quality rating BBN to be more construct-valid.

12.5.4 Equating Scaling

Scaling is a study of developing systematic rules and meaningful units of measurement for quantifying psychological constructs and empirical observations, for example, assigning IQ to measure intelligence. Equating is a study of establishing equivalent scores of two tests [45]. In the BBN constructed to obtain unit quality rating in this project, we equate MERLOT ratings with that of LORI, based on experts' opinions.

In this study, to achieve a rating-standard-neutral unit quality rating, we simply trust experts' input in uniting different rating standards. BBN is a means to interpret the unison structure. Thus, we could make use of all ratings that are available for recommendation, not only those that are rated by reviewers, but also those that are not rated but inferred from the existing rating information.

In short, the exact equating scaling process of MERLOT and LORI is not of interest in this work; rather, the applicability of using BBN is of interest. We use BBN and verify that the quality rating yielded from the constructed BBN conforms to that of the reviewers' using whichever rating standard, so that it facilitates learning object recommendation by sorting learning objects in quality at its own applied rating value.

The tests were run on simulated data. They are designed to determine whether, given realistic incomplete evaluation data, the model could make qualitatively plausible translations between instruments, the model could treat the reviews appropriately to their sources, and that these estimates increased in certainty as more data was acquired. We are able to confirm that the BBN upgrades or downgrades the quality rating in a plausible and meaningful fashion consistent with the graph topology and the NPTs, and that the certainty represented in the model increases as evaluation data accumulates.

12.5.5 Personalised and Collaborative Recommendation and Distribution of BBN

The BBN is used to facilitate rating frameworks that are personalized to the preferences of individual reviewers. Different personalized BBNs focusing on a single learning object can be combined to provide a singular quality rating for that object. We present the application of a polytree algorithm to resolve ratings across distributed BBN.

In this study, we do not differentiate to whom a learning object is recommended. All people evaluate all learning objects using one BBN; all users get learning object recommendations using one BBN. However, the role information is often quite important in educational practice. For example, for a disabled learner, accessibility is more important than it is for the others; or for a language learning object, interaction usability is more important than it is for a mathematics learning object. Therefore, there is a need for different quality rating BBN to be constructed to serve different purposes. In other words, we expect a system to automatically form personalized recommendations that account for the demonstrated preferences of a user and the requested types of learning object.

"We are on the verge of being able to provide learning customised for each specific learner at a specific time, taking into account, their learning styles, experience, knowledge and learning goals" [49]. After adaptive selection of appropriate objects based on individual needs, context is the second path for personalization of learning objects. The key for deploying learning objects effectively is to provide ways for the learner to contextualize the information [50].

One topic that appeared in our project discussion is to distribute a BBN model to a client computer, whose user is searching for a learning object. With the advancing technology in distributed computing, letting the client computer build a customized BBN on the fly is not unreasonable. We briefly evaluated a distributed belief network application, RISO [51]. This application distributes a BBN over a TCP/IP network using RMI[6] technology. Due to the time frame of this project, we did not pursue the topic any further; however, it is certainly worth looking at in detail.

Where distributing BBN and client-side BBN customization are technically possible, the challenge is, Do we allow client to change the topology of the distributed BBN? If yes, how does an on-line learning system take a customized BBN to make a personalized recommendation? Should we integrate the customized BBN into the master one that standard recommendation uses? If yes, then we will face the mathematically unsolved problem of multiple BBNs integration besides any other issue; if not, we will still face the computational difficulty of ensuring the topology validity of a customized BBN. BBN is an acyclic directed graph. To compute using a customized BBN, besides checking the nodes and NPT to be mutually exclusive and exhaustive, we have to constantly check whether any cycle exists in the BBN, which is computationally difficult. Additionally, without an educational expert intervention, it is difficult to ensure the content validity of a customized BBN, that is, the relationship between nodes.

We certainly have long way to go to realize personalization in learning object recommendation.

---

[6] RMI: remote method invocation. RMI is the Java version of what is generally known as a remote procedure call (RPC), in which objects on different computers can interact in a distributed network.

12.5.6 Share Learning Objects Among Multiple Repositories

As mentioned, BBN is able to conduct forward and backward inference. After a certain amount of observation, we will be able to correlate the quality rating from different rating standards, for example, between MERLOT and LORI. Thus, an adapter can be developed between two or more learning object repositories. When a learning object query is initiated, the result list can be returned from multiple learning object repositories. Based on a converted universal quality rating, this result list can be returned with recommendation. As such, we not only share the learning objects in a repository, but also access learning objects across different repositories. It expands the search base so that the user has more opportunity to access a better learning object.

In conjunction with network solutions such as eduSource Architecture [35], applying a rating standard adapter will likely accelerate sharing learning objects across repositories.

12.5.7 BBN Drawback

Having chosen BBN as the learning object recommendation technical strut, we recognize that there are a few drawbacks of BBN.

BBN is used to deal with uncertainties. Ideally, there should be some prior knowledge of the modeled domain, to form the unconditional event probability distribution. In the learning object quality rating BBNs used in this project, other than the nodes, other factors are all assumptions, including the parent to child node evenly distributed influence and the leaf parent node normal distribution, because we do not have any better knowledge or empirical study in this area. This affects the accuracy or certainty of the quality rating that we want to infer from these two BBNs.

To find a suitable, out-of-box BBN tool for this project remains a difficult task, since there are domain specific requirements. Another key issue is the computational performance of BBN. When users keep submitting reviews, how often should we propagate BBN probabilities so that the quality rating of learning objects can be updated without hindering the normal tasks of the on-line learning system?

An ideal system would take input and output scale fallacy into account. In this project, we have certain output scaling fallacy control, by weighing different evaluation differently based on reviewers. However, we do not have any mechanism to handle the input scale fallacy. In fact, for any recommendation system based on user input, there will always be certain degree of input scale fallacy.

12.5.8 Further Research Angles

There are a few other interesting questions that we would like to raise at this time. Currently we use normal distribution for leaf parent node, which gives us an ideal rating value inference; what if after sufficient experiments and going through the BBN justification process, we find out that leaf parent node does not have a normal distribution? In that case, would the rating value inference and the default value for unrated learning object still remain valid?

In integrated quality rating BBN, we use one node to represent expert panel rating. Ideally, a group of experts reaches a consensus after individual evaluation and group discussion, which can be taken into the BBN. However, in reality, experts might not agree with each other. In that case, should the BBN consider certain variance coefficient for this type of panel rating? If yes, how is it implemented?

In current learning object quality rating BBNs all nodes are atomic events. What if through the BBN learning process, we discover that one of the nodes has its own microstructure, for example, a BBN. How can this be implemented? How will this affect the recommendation performance?

We assume that the quality rating structure and its NPT are all discovered. The BBN constructed in this project mainly uses evenly distributed probabilities among all the parent nodes for one child node, based on the fact that there is no prior experience or better knowledge available in learning object rating. Moreover, these probabilities are a fixed set of values. Once the BBN is constructed, the probability distribution remains the same unless they are manually updated based on better understanding of the problem domain.

On the other hand, BBN can also be used to represent and reason about the task of learning the parameters, weights, and structure of each of these representations [46]. BBN is known for its ability to take evidence and discover the nodes and/or its conditional probability distribution, in other words, the self-learning capability. There are algorithms available to discover both nodes and the probability distribution table.

A popular heuristic approach to search hidden nodes is to use the hidden Markov model[7] [46]. This is a useful tool to verify current learning object quality rating standards and whether they have caught all variables in the quality evaluation model. More importantly, it can identify new variables, for example, the learning object aging issue. With the astonishing speed of human knowledge advancement, should we consider a chronological property for learning objects, at least in some categories, for example, information technology?

Another algorithm, EM (Expectation Maximization) [48], can find optimal conditional probability distribution of nodes, based on the evidence. In the context of learning object quality evaluation, this can provide a mechanism to automatically revise the BBN nodes' probability distribution after a certain number of reviews.

Another point to note is that BBN not only can work with discrete value, for example integer 1 to 5 in this project, but also can take or produce continuous value. We could explore and adjust current BBNs structure and interpolation techniques to obtain fine-grained rating values, like 2.34, 3.56, etc.

[7] The hidden Markov model is a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by transition probabilities. In a particular state, observation can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer. Therefore, states are "hidden" to the outside. [48]

These are promising techniques to enhance our current quality rating BBN. They ought to be explored further.

12.5.9 Conclusion

Our work explored a new way of obtaining the quality rating of learning objects. In applying Bayesian belief networks, a learning object repository is able to accept incomplete quality reviews. It also allows a learning object repository to obtain a standard neutral quality rating for learning objects, which facilitates sorting learning objects by quality upon learning object search request. Moreover, weight control over different types of evaluation increases the quality of the evaluations submitted to a learning object repository. The different types of evaluation include explicit and implicit ratings, as well as expert reviews and regular user reviews.

By proposing this new way of data measurement for learning object quality rating, it enhances the current learning object repository recommendation system, so that the entire on-line education efficiency can be improved.

Designers and developers of electronic learning today are being presented with a new content development landscape. Learning technology standards organisations are quickly moving toward open and industry-wide standards for learning objects. However, rather than preaching and waiting for conformance, we endeavor to create a mediator among technologies, owners, and users involved with learning objects. We seek an alternative way to satisfy a diversified on-line educational world.

Over the years it has become apparent that learning object repositories and the peripheral applications for learning object retrieval are of considerable international interest. Learning object recommendation is an inevitable course for on-line education to take. Our work is a small step toward a more effective on-line learning system. We expect the use of BBN to mature over time. We intend to provide richer case studies and examples and to continually reflect on our journey as we contribute a more intelligent learning object quality rating and recommendation system.

References

1. LTSC. (2000) Learning object. Learning technology standards committee Web site. ltsc.ieee.org.
2. Learning Object Metadata, LOM. (2000) http://ieeeltsc.org/wg12LOM/.
3. Fenton, N. (2000) What is BBN?, www.dcs.qmw.ac.uk/~norman/BBNs/BBNs.htm.
4. www.dei.isep.ipp.pt/docs/arpa.html.
5. Johnson, L.F. (2003) Elusive Vision: Challenges Impeding the Learning Object Economy, New Media Consortium.
6. Recker, M., Walker, A., Lawless, K. (2003) What do you recommend? Implementation and analyses of collaborative information filtering of web resources for education, Instructional Science, 31(4/5).

7. www.merlot.org.
8. www.elera.net/eLera/Home/About%20%20LORI/.
9. Nesbit, J.C., Belfer, K., Leacock, T. (2003) Learning Object Review Instrument User Manual version 1.5, eLera and POOL projects.
10. Rosenberg, M.J. (2001) E-Learning: Strategies for Delivering Knowledge in the Digital Age. New York: McGraw-Hill.
11. Friesen, N. (2004) Three objections to learning objects. In: Online Education Using Learning Objects. London: Routledge/Falmer, 2004.
12. ieeeltsc.org.
13. Wiley, D.A. (2002) Instructional use of learning objects. Agency for Instructional Technology.
14. cloe.on.ca.
15. Cisco Systems. (2003) Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches.
16. Hodgins, W. (2000) Into the Future: A Vision Paper. Technology and Adult Learning of the American Society for Training & Development, white paper.
17. Downes, S. (2001) Learning Objects: Resources for Distance Education Worldwide, International Review of Research in Open and Distance Learning, 2(1).
18. Wieseler, W. (1999) RIO: A standards-based approach for reusable information objects. Cisco Systems White Paper.
19. Longmire, W. (2000) Content and Context: Designing and Developing Learning Objects, Learning Without Limits, Volume 3: Emerging Strategies for e-Learning Solutions.
20. Downes, S. Learning object overview. http://www.learning-objects.net.
21. Marchionini, G. (1995) Information Seeking in Electronic Environments. Cambridge: Cambridge University Press.
22. Manning C., Raghavan, P. Text information retrieval, mining, and exploitation. www.stanford.edu/class/cs276b.
23. Levene, M. (2003) Recommendation system and collaborative filtering. www.dcs.bbk.ac.uk/~mark/download/lec7_collaborative_filtering.ppt.
24. Chesani, F. (2002) Recommendation systems. www-db.deis.unibo.it/courses/SI2/Relazioni/RecSystems.pdf.
25. Goldberg, D., Nichols, D., Oki, B.M., Terry, D. (1992) Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12):61–70.
26. Balabanovic, M., Shoham, Y. (1997) Fab: content-based collaborative recommendation. Communications of the ACM, 40(3):66–72.
27. Resnick, P., Lacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of ACM CSCW'94 Conference on Computer Supported Cooperative Work, pp. 175–186.
28. Melville, P., Mooney, R.J., Nagarajan, R. (2001) Content-boosted collaborative filtering. In: Proceedings of the ACM SIGIR-2001 Workshop on Recommender Systems, New Orleans, LA.
29. Pazzani, M.J. (1999) A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13(5–6):393–408.
30. Hill, J.R., Hannafin, J.R. (2001) Teaching and learning in digital environments: the resurgence of resource-based learning. Educational Technology Research and Development, 49(3):37–52.

31. Nesbit, J.C., Belfer, K., Vargo, J. (2002) A convergent participation model for evaluation of learning objects. Canadian Journal of Learning and Technology, 28(3).
32. Reiser, R.A., Kegelmann, H.W. (1994) Evaluating instructional software: a review and critique of current methods. Educational Technology Research and Development, 42(3).
33. Vargo, J., Nesbit, J.C., Belfer, K., Archambault, A. (2003) Learning object evaluation: computer-mediated collaboration and inter-rater reliability. International Journal of Computers and Applications. 25(3).
34. Hatala, M., Richards, G., Eap, T., Willms, J. (2004) The interoperability of learning object repositories and services: standards, implementations and lessons. WWW Conference, ACM.
35. Berger, J.O. (2000) Bayesian Analysis: A Look at Today and Thoughts of Tomorrow. JASA.
36. Berry, D.A. (1996) Statistics: A Bayesian Perspective. Duxbury Press.
37. Technical Report. (2002) Basics of Bayesian networks. www.agena.co.uk.
38. Pearl, J. (1998) Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann.
39. www.hugin.com.
40. plato.stanford.edu/entries/bayes-theorem.
41. Walsh, W.B., Betz, N.E. (1995) Tests and Assessment. 4th ed., p. 49, Prentice Hall.
42. Walsh, W.B., Betz, N.E. (1990) Tests and Assessment. 4th ed., p. 58, Prentice Hall.
43. Hunter, J.E., Schmidt, F.L. (1990) Methods of Meta-Analysis: Correcting Error and Bias in Research Findings. Newsbury Park: Sage Publications.
44. Crocker, L., Algina, J. (1986) Introduction to Classical and Modern Test Theory. Forth Worth, TX: Harcourt Brace Jovanovich College Publishers.
45. Buntine, W.L. (1994) Operations for learning with graphical models. Journal of Artificial Intelligence Research, 2:159–225.
46. Rabiner, L.R., Juang, B.H. (1986) An Introduction to Hidden Markov Models. IEEE ASSP Magazine, January.
47. Dempster, A., Laird, N., Rubin, D. (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 39(B).
48. Schatz, S. (2001) Paradigm Shifts and Challenges for Instructional Designers.
49. Longmire, W. (2000) A Primer on Learning Objects. Learning Circuits, March.
50. Dodier, R. (1999) Unified Prediction and Diagnosis in Engineering Systems by Means of Distributed Belief Networks. PhD Dissertation. Dept. of Civil, Environmental, and Architectural Engineering, University of Colorado.
51. www.norsys.com/index.html.
52. Kumar V., Groeneboer C., Chu, S. (2005) Sustainable learning ecosystem. Tutorial, International Conference on Advanced Learning Technologies, ICALT 2005, Kaohsiung, Taiwan.
53. Hatala, M., Richards, G., Eap, T., Willms, J. (2004) eduSource: Implementing Open Network for Learning Repositories and Services. Special Track on Engineering e-Learning Systems held at ACM Symposium on Applied Computing (SAC).
54. www-2.cs.cmu.edu/~javabayes/Home/.
55. CanCore. (2003) CanCore guidelines version 2.0: educational category. www. cancore.ca.
56. Carey , T., Swallow, J., Oldfield, W. (2002) Educational rationale metadata for learning objects. Canadian Journal of Learning and Technology, 28(3):55–71.
57. Mager, R. (1975) Preparing Instructional Objectives, 2nd ed. Palo Alto, CA: Fearon.
58. Mwanza, D., Engestrom, Y. (2005) Managing content in e-learning environments. British Journal of Educational Technology, 36:453–463.

59. Greenberg, J. (ed.). (2000) Metadata and Organizing Educational Resources on the Internet. Binghamton, NY: Haworth.
60. IEEE-LTSC. (2002) IEEE Standard for Learning Object Metadata (1484.12.1-2002).
61. IMS Digital Repositories Interoperability: Core Functions Information Model, Version 1. www.imsglobal.org/digitalrepositories/index.cfm.