

行政院國家科學委員會專題研究計畫 成果報告

分類技術與貝氏網路之應用：法學文件之語意標記與人機互動之使用者建模

計畫類別：個別型計畫

計畫編號：NSC94-2213-E-004-008-

執行期間：94年08月01日至95年10月31日

執行單位：國立政治大學資訊科學系

計畫主持人：劉昭麟

計畫參與人員：黃珮雯、鄭人豪、陳禹勳及林仁祥

報告類型：精簡報告

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中 華 民 國 95 年 10 月 4 日

行政院國家科學委員會補助專題研究計畫 出國報告

分類技術與貝氏網路之應用： 法學文件之語意標記與人機互動之使用

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 98-2213-E-004-008

執行期間： 94 年 8 月 1 日至 95 年 10 月 31 日

計畫主持人：劉昭麟

共同主持人：

計畫參與人員：黃珮雯、鄭人豪、陳禹勳及林仁祥

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立政治大學 資訊科學系

中 華 民 國 95 年 10 月 4 日

Abstract

We report the research work for investigating the annotation of judicial documents in Chinese and applying Bayesian networks for student modeling. This piece of work embarked in the year 2005 and will continue toward 2008. We have achieved reasonable results in the first year.

Overview

In the past many years, we have studied classification techniques for categorizing judicial documents in Chinese. The categorization of judicial documents can be useful in practice if we can achieve satisfactory accuracy. Although we hope, the actual application of our system may not take place in the courts. With our current achievements, we see that we can build a Google-like server for judicial consultation. The main difference between our system and Google will be that we do not require users to choose and type in key words for search. In a normal prosecution procedure, the defendant will receive a prosecution document from the courts. To know that the prosecution documents are about, the users just feed the whole file to our system to search similar prior documents. Our system can provide prior documents that are similar to the current document based on prosecution reasons or legal articles that might be cited for the current case.

In addition to the categorization of judicial documents, we have also attempted to apply machine learning-based methods for learning student models. The input data to our learners are simulated students' records for taking tests. We implemented the simulator in the past year, and are continuing to improve it. Given students' test records, our classifier try to tell how students learning composite concepts. We have identified some key issues in this research direction, and expect to work on them in the continued projects. At this moment, we see that there are chances that computers can help educational experts to select detailed models about students' learning patterns. However, this is not a simple work, particularly when students' test records do not deterministically reflect students' competence.

Technical skeletons

Since we have applied very different approaches for the document classification and student modeling problems in our research, we have to provide the skeletons separately.

Classification of judicial documents

As in the work that we did in the past many years, we applied k nearest neighbor (k NN) methods for the classification task. With a preprocessing procedure, we extracted key information from the documents and converted them into a set of features. In applying k NNs, we calculated the similarity between two documents with the similarity measure defined based on the feature sets.

In the past, we have been using word-based features in our work. A Chinese word is a sequence of characters that we segmented from a normal Chinese text. In our research that took place between 2004 and 2005, we have applied the introspective learning method to adjust the weights for the keywords, hoping to improve the accuracy of our classifiers.

This year, we switched to phrase-based features. The hunch is that using phrases, consisting of two words, should make the phrase more specific in their semantics, and hopefully can

improve the effectiveness of our classifiers. The creation and weighting of the phrases and the evaluation of our methods had been reported in an international conference. Please be referred to the appended paper for more details.

Student modeling with Bayesian networks

We believe that what reported in this summary is a brand new issue that one can find in the literature. We will make a great contribution to the world, if this research direction eventually leads to real world applications.

How do we know how students learn composite concepts? When a composite concept consists of multiple basic concepts, there can be many different ways to learn it. For instance, there are at least 14 different ways to learn a composite concept that contains four basic concepts. (Please see the appended papers for reasons.) A human teacher may believe that s/he knows how her/his students learn. However, such beliefs are generally not critically verified. We do not intend to disregard human intuition, but we believe that machines can be useful in searching for the real learning process.

We organize our work into several components. Since this is the first step of our study, we do not have data for real students yet. Instead, we implemented a student simulator that is structurally similar to a computer-assisted student assessment system. The simulator can generate students' test records, which will be used in the place of test records of real students.

Given the student records, we employed several classification techniques to guess the learning patterns. In running the simulator, we had to provide key information about students' learning patterns so that the simulator can create test records accordingly. Such key information was known to us but was not provided to our classifiers. Hence, our classifiers had to guess these hidden learning patterns.

The details of how we conducted the experiments are provided in the appended papers. It is found that our classifiers can hit the current answer, if the experimental settings are favorable. However, the problem is not as easy as it may appear, and our classifiers performed not clearly better than a random guesser when the settings are really unfavorable.

Published papers

Since we have conducted quite a lot of work in a year, we thought it might be more direct to provide the papers that we published and presented in international conferences for both the NSC reviewers and the ordinary public to know more about our achievements.

We have published our papers in AI and IEEE conferences. We provided the list, and the papers follow.

- C.-L. Liu. Learning students' learning patterns with support vector machines, Lecture Notes in Computer Science 4203: Proceedings of the Sixteenth International Symposium on Methodologies for Intelligent Systems (ISMIS'06), 601-611. Bari, Bari, Italy, 27-29 September 2006. (SCIE)
- C.-L. Liu and C.-D. Hsieh. Exploring phrase-based classification of judicial documents for criminal charges in Chinese, Lecture Notes in Computer Science 4203: Proceedings of the Sixteenth International Symposium on Methodologies for Intelligent Systems (ISMIS'06), 681-690. Bari, Bari, Italy, 27-29 September 2006. (SCIE)
- C.-L. Liu and Y.-T. Wang. An experience in learning about learning composite concepts,

Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06), 187-189. Kerkrade, Limburg, Netherlands, 5-7 July 2006. (EI?)

- C.-L. Liu. Learning how students learn with Bayes nets, Lecture Notes in Computer Science 4053: Proceedings of the Eighth International Conference on Intelligent Tutoring Systems (ITS'06), 772-774. Jhongli, Taiwan, 26-30 June 2006. (SCIE)

Learning Students' Learning Patterns with Support Vector Machines

Chao-Lin Liu

Department of Computer Science, National Chengchi University, Taiwan
chaolin@nccu.edu.tw

Abstract. Using Bayesian networks as the representation language for student modeling has become a common practice. Many computer-assisted learning systems rely exclusively on human experts to provide information for constructing the network structures, however. We explore the possibility of applying mutual information-based heuristics and support vector machines to learn how students learn composite concepts, based on students' item responses to test items. The problem is challenging because it is well known that students' performances in taking tests do not reflect their competences faithfully. Experimental results indicate that the difficulty of identifying the true learning patterns varies with the degree of uncertainty in the relationship between students' performances in tests and their abilities in concepts. When the degree of uncertainty is moderate, it is possible to infer the unobservable learning patterns from students' external performances with computational techniques.

1 Introduction

Providing satisfactory interaction between human and machines requires good computational models of human behaviors. Take computer-adaptive testing (CAT) for example. Researchers build models based on the Item-Response Theory (IRT) [1,2] and Concept Maps [3] for predicting students' performances and selecting appropriate test items for assessment. With good student models, a computational system can evaluate students' competence with less test items than traditional paper-and-pencil tests will need, and can achieve better accuracy in its evaluation. In addition, test takers can access a CAT system almost any time at any location, and can obtain their scores on the spot. Hence, CAT has been adopted in many official evaluation activities, including TOFEL and GRE, although there are sporadic criticisms [4].

Typically, domain experts provide information about student models, which are then implemented with computational techniques. CAT systems that adopt IRT assume that a student's responses to test items are mutually independent given the student's competence, so IRT-based systems generally take the so-called naïve Bayes models [5, 6]. Based on this assumption, the problem of building student models boils down to learning the model parameters from observed data [7, 8]. Similarly, Liu et al. assume the availability of concept maps of students and teachers, and design algorithms for comparing the concept maps for assessment [3].

Although human experts can choose good models from candidate models, they may not agree on their choices. For instance, Millán and Pérez-de-la-Cruz discussed a hierarchical structure of Bayesian networks [9] that included nodes for *subjects*, *topics*, *concepts*, and *questions* [10]. Vomlel employed nodes for *skills* and *misconceptions*, and used relevant nodes as direct parents of nodes for *tasks* [11].

In this paper, we explore computational techniques for comparing the candidate models for students. Although we do not expect computational techniques will give better model structures than human experts will do in the short term, we hope that computational techniques can assist human experts to identify more precise models. More specifically, we would like to guess how students learn composite concepts. A *composite* concept results from students' integration of multiple *basic* concepts. Let $dABC$ denote the composite concept that involves three basic concepts cA , cB , and cC . How do we know how students learn the composite concept? Do they learn $dABC$ by directly integrating the three basic concepts, or do they first integrate cA and cB into an intermediate product, say dAB , and then integrate dAB with cC ?

We compare candidate models that are represented with Bayesian networks, based on students' responses to test items. Students' responses to test items reflect their competences in the tested concepts in an indirect and uncertain manner. The relationship is uncertain because students may make inadvertent errors and luckily hit the correct answers. We refer to these situations as *slip* and *guess*, respectively, henceforth. *Slip* and *guess* are frequently cited in the literature, and many researchers adopted Bayesian networks to capture the uncertainty in their CAT system, e.g., [6, 8, 10-12].

As a result, our target problem is an instance of learning Bayesian networks. This is not a new research problem, and a good tutorial is already available [13]. However, learning Bayesian networks for student modeling is relatively rare, based on our knowledge, particularly when we would try to induce a network directly from students' item responses. Vomlel created network structures from students' data and applied principles provided by experts to refine the structures [11]. Besides those difficulties for learning structures from data, learning a Bayesian network from students' data is more difficult because most of the variables of interests are not directly observable. Hence, the problem involves not just missing values and not just one or two hidden variables.

In our experiments, we have 15 basic and composite concepts. We cannot observe whether students are competent in these concepts directly, though we assume that we can collect students' responses to test items that are related to these concepts. The problem of determining how students learn composite concepts is equivalent to learning the structure of the hidden variables given students' item responses.

We propose mutual information (MI) [14] based heuristics, and apply the heuristics for predicting the hidden structures in two ways: a direct application and training support vector machines (SVMs) [15] for the prediction task. Experimental results indicate that it is possible to figure out the hidden structures under moderate uncertainty between students' item responses and students' competence.

We provide more background information in Section 2, introduce the MI-based heuristics in Section 3, present the SVM-based method in Section 4, and wrap up this paper with a discussion in Section 5.

2 Preliminaries

We provide more formal definitions, explain the source of the simulated students' item responses, and analyze the difficulties of the target task in this section.

Table 1. One of the Q-matrices used in the experiments

group	cA	cB	cC	cD	dAB	dAC	dAD	dB	dB	dCD	dABC	dABD	dACD	dBCD	dABCD
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1
3	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1
4	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1
5	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1
6	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1
7	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1
8	1	1	1	1	1	0	0	0	0	1	1	0	0	0	1
9	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1
10	1	1	1	1	1	0	0	0	0	1	0	1	1	0	1
11	1	1	1	1	1	1	0	0	1	1	0	1	0	1	1
12	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1
13	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1
14	1	1	1	1	1	0	0	0	0	1	0	0	1	0	1
15	1	1	1	1	0	0	0	0	1	1	0	0	0	1	1
16	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1

The goal of our work is to find the hidden structure of the unobservable nodes that represent students' competence in concepts, based on observed students' item responses. We assume that students learn composite concepts from *parent concepts* that do not have overlapping basic concepts. For the problem of investigating how students learn $dABC$ that we mentioned in Section 1, we assume that there are four possible answers: AB_C , AC_B , BC_A , and A_B_C , where the underscores separate the parent concepts. Although there is no good reason to exclude a learning pattern like AB_BC , including such overlapping parent concepts will dramatically make the problem more complex. We obtained students' item responses from a simulation program that was reported in a previous work [6]. In this paper, we will try to learn how students learn $dABCD$, and there are 14 possible ways to learn this target concept.

2.1 Creating Simulated Students

Although not using real students' data subjects our work to criticisms, we believe that, if we can employ computational models to predict students' behaviors in CAT systems, we should believe that the same computational model is trustworthy for simulating students' behaviors. The use of simulated students is not our invention, previous and well-known work has taken the same approach for studying computational methods, e.g., [10, 12].

Using Liu's simulator that is described in [6], we can control the structure of the Bayesian network and the generation of the conditional probability tables (CPTs). The generation of the CPTs relies on a random number generator that uniformly samples numbers from a given range. In order to specify competence patterns of the student population, we also have to provide a matrix that is similar to the Q-matrix [16], and Table 1 shows the matrix that we used in many of our experiments. The columns are concepts, and the rows are student groups. When the column is a basic concept, cells are 1 if a typical student in the student group is competent in the concept, and cells are 0 otherwise. When the column is a composite concept, cells are 1 if a typical student in the student group is able to integrate the parent concepts to form the composite concept if s/he is competent in the parent concepts. Although the cells are either 0 or 1, Liu employed random numbers to give an uncertain relationship between

student groups and competence in concepts through a simulation parameter: *groupInfluence*. A student's behavior may deviate from his/her typical group competence pattern with a probability that is uniformly sampled from the range $[0, \textit{groupInfluence}]$.

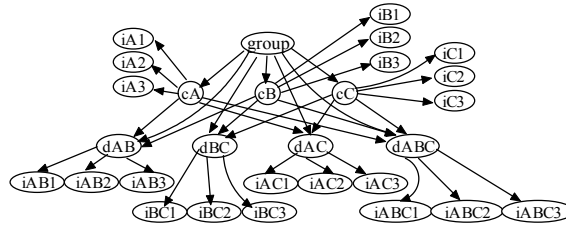


Figure 1. A Bayesian network for 3 basic concepts

We assumed that every concept had three test items in the experiments, and the network shown in Figure 1 shows a possible network when we consider the problem in which there are only three basic concepts. Note that in this network, we assume that students learn *dABC* by directly integrating the three basic concepts, which is indicated by the direct links from the basic concepts to the node labeled *dABC*. We cannot show the network for the case in which there are four basic concepts in this paper, due to the size of the network. (There will be 15 nodes for concepts, 3×15 nodes for test items, and a lot more links between these 60 nodes.)

The probabilities of *slip* and *guess* are also controlled by a simulation parameter: *fuzziness*. Students of a student group may deviate from the typical behavior with a probability that is uniformly sampled from $[0, \textit{fuzziness}]$.

Given the network structure, the Q-matrix, and the simulation parameters, we can create simulated students. In our experiments, we assumed that a student can belong to any of the 16 student groups with equal probabilities. Following Liu's strategy, we used a random number, ρ that was sampled from $[0, 1]$ to determine whether a student would respond to a test item correctly or incorrectly. The conditional probability of correctly responding to a test item given a student belonged to a particular group can be calculated easily with Bayesian networks. Consider the instance for *iA1*, a test item for *cA*. If ρ is smaller than $\Pr(iA1 = \textit{correct} \mid \textit{group} = g_1)$ when we simulated a student who belonged to the first group, we assumed that this student responded to *iA1* correctly. Since there were 15 concepts, a record for a simulated student would contain the correctness for each of 45 ($=3 \times 15$) test items.

After using the networks to create simulated students, we hid the networks from our programs that took as input the item responses and guessed the structures of the hidden networks.

2.2 Contents of the Q-Matrix and Problem Complexity

The contents of the Q-matrix influence the prior distributions of students' competence patterns and the performance of simulated students [12]. Clearly, there can be many different ways to set the contents of the matrix.

We set the Q-matrix in Table 1, partially based on our experience. Notice that all columns for the basic concepts and the target concept, *dABCD*, are 1. This should be considered a normal choice. If we do want to learn how students learn *dABCD*, we should try to recruit students who appear to be competent in *dABCD* to participate in our experiments. In addition, there is no good reason to recruit anyone who is not competent in any of the basic concepts in the experiments. We set the values for *dABC*, *dABD*, *dACD*, and *dBCD* to 16 possible combinations, and this is why we

include 16 student groups in Table 1. We randomly choose the values of dXY , where X and Y are symbols for basic concepts, and will report experimental results for other possible settings.

When we consider β basic concepts in the problem, the number of possible ways to learn how students learn a composite concept that is comprised of all these β concepts is related to the Stirling number of the second kind [17]. It is easy to verify that this number grows rapidly with β , and we will have 14 alternatives if we set β to 4.

$$\sum_{i=2}^{\beta} \left(\frac{1}{i!} \sum_{j=0}^{i-1} (-1)^j \binom{i}{j} (i-j)^{\beta} \right) \quad (1)$$

3 MI-Based Heuristics

Consider the situation when we generate students' data from the network shown in Figure 1. If we have the true states of all the concept nodes, it is not difficult to learn the network structure with a variant of the PC algorithm [18] implemented in Hugin [19]. However, we cannot observe the true competence levels of students in reality, and can only indirectly measure the competence through the results of examinations. Moreover, the item responses do not perfectly reflect students' competence, due to many reasons including *guess* and *slip*. Hence, we need to find indirect evidence that may help us to predict the hidden structure.

3.1 Estimating the MI Measures

Recall that we have assumed that every simulated student will respond to three test items for each concept. Out of three test items, a student may correctly respond to 0%, 33%, 67%, and 100% of the test items. Hence, it is possible to estimate the state of the concept nodes with the percentage of correct responses. We can also use the percentages to estimate the state of a set of variables. For instance, $\Pr(cA=33\%, cB=66\%)$ can be the percentage of students who correctly respond to exactly one item for cA and two items for cB . Given such estimates, we will be able to compute the mutual information between any two sets of variables, and apply the estimated mutual information in guessing the hidden structure.

Intuitively, the variables that are more closely related to each other will exhibit higher mutual information. Partial networks shown in Figure 2 include five possible ways to learn $dABCD$, i.e., A_B_CD ,

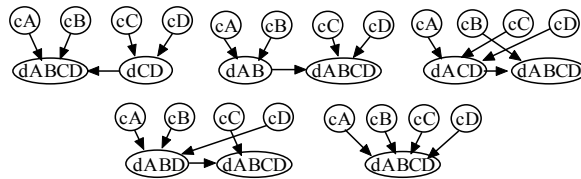


Figure 2. Candidate (partial) Bayesian networks in our experiments

AB_C_D , ACD_B , ABD_C , and $A_B_C_D$. Let $MI(X;Y)$ denote the mutual information between two sets of variables, X and Y . If A_B_CD is the true structure, we expect that it is more likely for the estimated $MI(cA, cB, dCD; dABCD)$ to be larger than the estimated $MI(dAB, cC, cD; dABCD)$ and other estimated MI measures. Hence, we can employ the following heuristics.

Heuristics: The structure that has the largest estimated MI measure is the hidden structure.

Before we estimate the MI measures, we add 0.001 to the number of occurrences of every possible combination of variables. This will avoid the zero probability problems, and is a typical smoothing procedure for estimating probability values [20].

3.2 Experimental Evaluation

Figure 3 shows the flow of how we evaluated the heuristic. In the experi-

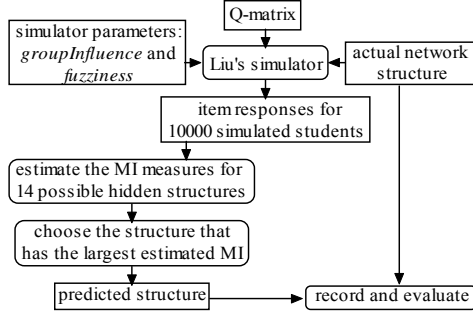


Figure 3. Flow for evaluating the heuristic

ments, we used five different network structures to create simulated students, and their main differences are shown in Figure 2. The parent concepts of other composite concepts that do not appear in the sub-networks in Figure 2 are the basic concepts. For instance, the parent concepts of *dABD* in the network that used the leftmost sub-network in the top row of Figure 2 are *cA*, *cB*, and *cD*. As we mentioned in Section 2, we mainly used the Q-matrix in Table 1 in our experiments. We set *groupInfluence* and *fuzziness* to different values in $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$, so there were 36 combinations. We did not try values larger than 0.3 because they were beyond consideration normally discussed in the literature. For each of the five network structures and a combination of *groupInfluence* and *fuzziness*, we sampled 600 network instances with the Q-matrix shown in Table 1, and created a different population of 10000 simulated students for each of these instances. The choice of “10000” was arbitrary, and the goal was to make each of the 16 groups include many students.

An *experiment* corresponded to a different combination of *groupInfluence* and *fuzziness*, so there were 36 experiments. We used *accuracy* to measure the quality of our prediction of the hidden structures. It was defined as the percentage of correct prediction of 3000 ($=5 \times 600$) randomly sampled network instances that were used to create the simulated students. According to Equation (1), there were 14 possible answers when β is 4. Hence, to guess the hidden structure of each of these 3000 network instances, we calculated the estimated MI measures for 14 possible answers from the item responses of the 10000 simulated students.

Figure 4 summarizes the experimental results. The vertical axis shows the accuracy, the horizontal axis shows the decimal part of *fuzziness*, and the legends mark the values of *groupInfluence* used in the experiments. Curves in these charts show a general trend that we expected. Increasing the values of *groupInfluence* and *fuzziness* made the relationship between students’ item responses and their competence patterns more uncertain and our prediction less accurate. When

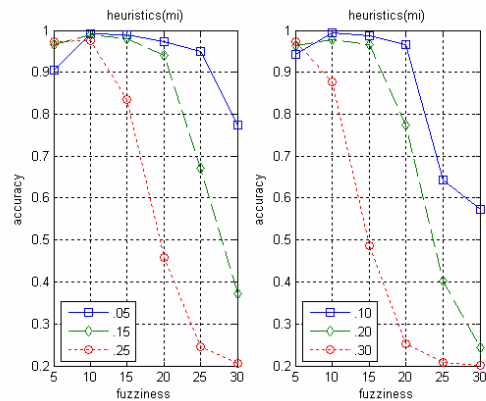


Figure 4. Accuracy achieved by the MI-based heuristics

both *groupInfluence* and *fuzziness* were both close to 0.3, the accuracy was about 0.2.

It is easy to interpret 0.2 as a result of random guesses from five possible answers, but this is not correct. Although we used only five network structures that are shown in Figure 2 to create simulated students, our prediction program did not take this into account, and could consider network structures that were not included in Figure 2. The fact is that our heuristics favored particular structures, which we learned by looking into the internal data collected in experiments. When both *groupInfluence* and *fuzziness* were large, the heuristics tended to favor *A_B_C_D*, which happened to be one of the true answers. As a result, we had the accuracy of 0.2. Had we excluded *A_B_C_D* from the true networks, the accuracy would become smaller than 0.2.

Although we expected that the accuracy should improve as we reduced the values of *groupInfluence* and *fuzziness*, the experimental results did not fully support this intuition. (When we conducted experiments for the cases in which there were only three basic concepts, experimental results did support this intuitive expectation.) When both *groupInfluence* and *fuzziness* were close to 0.05, the heuristics tended to favor *AB_CD* against other competing structures, making the accuracy worse than we expected. More specifically, we created a 14×14 confusion matrix [20] and found that our heuristics chose *AB_CD* relatively frequently when the true structures were *A_B_CD* and *AB_C_D*. The accuracy could hit as low as 0.85 when both *groupInfluence* and *fuzziness* were both 0.05 for other Q-matrices that were different from the Q-matrix shown in Table 1. These Q-matrices were different in the settings in the *dXY* columns, where both *X* and *Y* represents a basic concept. This phenomenon is certainly not desirable, though understandable, because of the similarity between and *A_B_CD*, *AB_C_D*, and *AB_CD*.

When the heuristics led us to choose a wrong structure, the estimated MI measures for the chosen structures were not larger than the MI measures for the correct structures by a big margin. In fact, we found that, when the heuristics failed to choose the correct answers, most of the estimated MI measures were very close to each other. Hence, we expect that if we consider the ratios between the estimated MI measures, we may design more effective heuristics. We included ratios between estimated measures as features for building the SVM-based classifiers that we report below.

4 SVM-based Methods

Support vector machines [15] are a relatively new formalism that can be applied to the task of classifications. We can train SVMs with training patterns that are associated with known class labels, and the trained

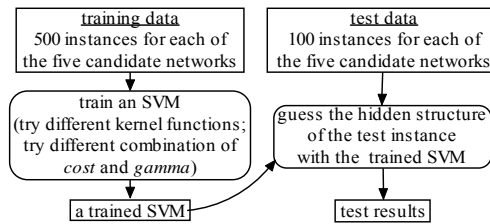


Figure 5. Guessing the hidden structure with SVMs

SVMs can be used to predict the classes of test patterns. In this work, we employed the LIBSVM packages provided by Chang and Lin [21].

4.1 Preparing for Experiments

Figure 5 summarizes the main steps that we took to apply SVMs in our work. As explained in Section 3.2, we obtained students' data in 36 different experiments. In

each of these experiments, there were 600 network instances for each of the candidate networks shown in Figure 2. Therefore, we used students' data obtained from 500 network instances for each of the candidate network as the training data, and used the students' data obtained from the remaining 100 network instances as the test data.

Figure 6 summarizes how we prepared the training and test instances. In addition to the original 14 estimated MI measures, we also computed ratios between the estimated MI measures as features. The introduction of ratios was inspired by analyses that we discussed at the end of Section 3.2. We divided the original 14 estimated MI measures by the largest estimated MI measure in each training instance.

This gave us 14 new features. We also divided the largest estimated MI measure by the second largest estimated MI measure, and divided the largest estimated MI measure by the average of all estimated MI measures. This gave us 2 more features, so we used 30 features for each of the 500 training instances for each of the five candidate networks. The true answers (also called *class labels*) were attached to the instances for both training and testing. In summary, we created a training instance from 10000 simulated students, and there were 2500 (=5×500) training instances, each with 30 attributes and a class label. When testing the trained SVMs, we produced the 16 extra features from the original 14 estimated MI measures for each of the test instances as well. The true answer was attached to the test instance so that we could compare the true and predicted answers, but the SVMs did not peek at the true answers.

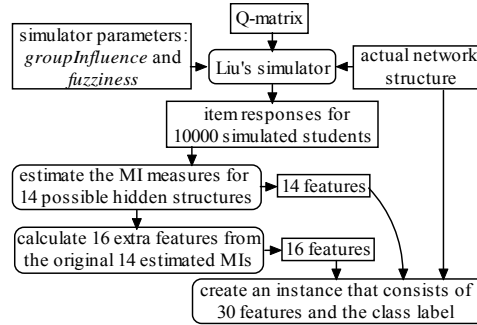


Figure 6. Preparation of the training and test data

4.2 Results

Charts shown in Figure 7 show the experimental results. The vertical axis, the horizontal axis, and the legend carry the same meanings as those for charts in Figure 4. The titles of the charts indicate what types of SVMs we used in the experiments. We used the c-SVC type of SVMs in all experiments, and tried three different kernel functions, including polynomial (c-svm-poly), radial basis (c-svm-rb), and sigmoid (c-svm-sm) kernels. Among these tests, using polynomial and radial basis kernels gave almost the same accuracy, and both performed better than the sigmoid kernel. However, it took a longer time for us to train an SVM when we used the polynomial kernel.

Comparing the curves for the same experiments in Figures 3 and 4 show the significant improve-

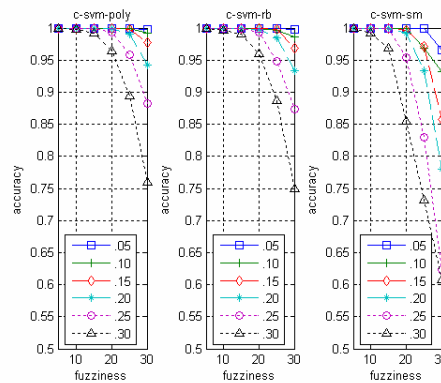


Figure 7. Accuracy achieved by the SVM-based methods

ments achieved by using the SVMs. In the middle chart in Figure 7, the accuracy stays above 0.75 even when *groupInfluence* and *fuzziness* were 0.3. The heuristics-based method got only 0.2 in accuracy under the same situation. In addition, the trends of all curves support our intuitive expectation—larger *groupInfluence* and *fuzziness* would lead to worse accuracy. The problem that occurred in the upper left corners of the charts in Figure 4 was also gone. Even when we tried different Q-matrices, smaller *groupInfluence* and *fuzziness* also made the prediction of the hidden structure easier than when we used larger *groupInfluence* and *fuzziness*.

We have to explain that we had to search for the best parameters for SVMs when we trained SVMs. In particular, we ran experiments that used different values for *cost* and *gamma* in LIBSVM, using default values for other parameters. Different combinations of *cost* and *gamma* led to different accuracy in guessing the hidden structures for the test data. In our experiments we tried combinations of *cost* and *gamma* from values in $\{0.1, 0.2, \dots, 1.9\}$, and used the best accuracy for the test data in 381 ($=19 \times 19$) cases when we prepared charts in Figure 7.

5 Concluding Remarks

We tackle a student modeling problem that requires us to infer the hidden model for learning composite concepts, based on observations of variables that have only indirect and uncertain relationships with variables in the hidden model. Experimental results indicate that this task is not impossible, and we can actually achieve good results when the situations are favorable.

We report results of two different approach—A heuristics-based and an SVM-based approach. Charts shown in Figures 3 and 7 clearly show that the SVM-based approach is more effective. However, the advantages of our SVM-based approach come at some costs. We will need experts to enumerate the candidate networks and guess the contents of the Q-matrix. Only after obtaining these information, can we create simulated students and train the SVMs, which will then be used to guess the hidden structure with students' item responses. (Although we did not use item responses of real students in our experiments, we will have to do so in reality.)

Though we cannot discuss all experimental results in this paper, our experience indicates that the contents of the Q-matrix affect the final accuracy. The contents of the Q-matrix show what types of students that we should recruit for investigating how the students learn the composite concepts. Results reported in this paper were acquired with the Q-matrices that assumed students were competent in *dABCD* and all basic concepts. If we allow cells in these columns to be zero, then the final accuracy will be affected. However, we do not think this should happen in reality. If we do want to learn how students learn *dABCD*, we should have tried as hard as possible to collect item responses from students who appear to be competent in *dABCD* and all basic concepts. Given the intentionally introduced uncertainties, i.e., *groupInfluence* and *fuzziness*, our algorithm does allow errors in recruiting students, and experts do not have to provide very exact information about the Q-matrices. The SVM-based approach is reasonably robust in this aspect.

We hope the reported results can be useful for student modeling in reality. Our experience underscores the importance of experts' opinion for the success of the modeling task. Experimental results also show the potential applicability of the heuristics-

based methods for selecting the correct hidden sub-structure even when experts' opinions were not available.

Acknowledgements

This research was partially supported by contract NSC-94-2213-E-004-008 of the National Science Council of Taiwan. We gratefully thank the reviewers for their invaluable comments, and will answer their questions that we cannot do so in this page-limited paper during the oral presentation.

References

1. I. W. J. van der Linden and C. A. W. Glas, *Computerized Adaptive Testing : Theory and Practice*, Kluwer, Dordrecht, Netherlands, 2000.
2. H. Wainer et al., *Computer Adaptive Testing : A Primer*, Lawrence Erlbaum Associates, NJ, USA, 2000.
3. C.-C. Liu, P.-H. Don, and C.-M. Tsai, "Assessment based on linkage patterns in concept maps," *J. of Information Science and Engineering*, vol. 21, pp. 873–890, 2005.
4. G. Smith, "Does Computer-Adaptive Testing Make the Grade?" ABCNEWS.com, 17 March 2003.
5. R. J. Mislevy and R. G. Almond, "Graphical models and computerized adaptive testing," CSE Technical Report 434, CRESST/Educational Testing Services, NJ, USA, 1997.
6. C.-L. Liu, "Using mutual information for adaptive item comparison and student assessment," *J. of Educational Technology & Society*, vol. 8, no. 4, pp. 100–119, 2005.
7. F. B. Baker, *Item Response Theory : Parameter Estimation Techniques*, Marcel Dekker, NY, USA, 1992.
8. R. J. Mislevy, R. G. Almond, D. Yan, and L. S. Steinberg, "Bayes nets in educational assessment: Where do the numbers come from?" in *Proc. of the Fifteenth Conf. on Uncertainty in Artificial Intelligence*, pp. 437–446, 1999.
9. J. Pearl, *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*, Morgan Kaufmann, CA, USA, 1988.
10. E. Millán and J. L. Pérez-de-la-Cruz, "A Bayesian diagnostic algorithm for student modeling and its evaluation," *User Modeling and User-Adapted Interaction*, vol. 12, no. 2-3, pp. 281–330, 2002.
11. J. Vomlel, "Bayesian networks in educational testing," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 12, no. Supplement 1, pp. 83–100, 2004.
12. K. VanLehn, Z. Niu, S. Siler, and A. Gertner, "Student modeling from conventional test data : A Bayesian approach without priors," *Lecture Notes in Computer Science*, vol. 1452, pp. 434–443, 1998.
13. D. Heckerman, "A tutorial on learning with Bayesian networks," in M. I. Jordan (ed.), *Learning in Graphical Models*, pp. 301–355, MIT Press, MA, USA, 1999.
14. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, NY, USA, 1991.
15. C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, pp. 273–297, 1995.
16. K. K. Tatsuoka, "Toward an integration of item-response theory and cognitive error diagnoses," in N. Fredericksen et al. (eds.), *Diagnostic Monitoring of Skill and Knowledge Acquisition*, Erlbaum, NJ, USA, 1990.
17. D. E. Knuth, *The Art of Computer Programming: Fundamental Algorithms*, Addison-Wesley, MA, USA, 1973.
18. P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, second edition, MIT Press, MA, USA, 2000.
19. Hugin: <http://www.hugin.com>
20. I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, second edition, Morgan Kaufmann, CA, USA, 2005.
21. C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
22. MATLAB: <http://www.mathworks.com>

Exploring Phrase-Based Classification of Judicial Documents for Criminal Charges in Chinese

Chao-Lin Liu and Chwen-Dar Hsieh

Department of Computer Science, National Chengchi University, Taiwan
chaolin@nccu.edu.tw

Abstract. Phrases provide a better foundation for indexing and retrieving documents than individual words. Constituents of phrases make other component words in the phrase less ambiguous than when the words appear separately. Intuitively, classifiers that employ phrases for indexing should perform better than those that use words. Although pioneers have explored the possibility of indexing English documents decades ago, there are relatively fewer similar attempts for Chinese documents, partially because segmenting Chinese text into words correctly is not easy already. We build a domain dependent word list with the help of Chien's PAT tree-based method and HowNet, and use the resulting word list for defining relevant phrases for classifying Chinese judicial documents. Experimental results indicate that using phrases for indexing indeed allows us to classify judicial documents that are closely similar to each other. With a relatively more efficient algorithm, our classifier offers better performances than those reported in related works.

1 Introduction

We investigate the effectiveness of applying phrases for indexing judicial documents in Chinese. Conventional wisdom and experimental results suggest that phrases provide better indications of contents of the indexed documents than keywords, thereby offering better chances of higher quality of information retrieval. Indeed, many natural languages contain homonyms and polysemes, so using isolated keywords for indexing takes the risk of interpreting words as unintended senses, and using phrases helps to alleviate the ambiguity problems with the contextual information provided by the surrounding words. Due to this intuition, Salton, Yang and Yu have pioneered the applications of phrases for indexing English documents as early as 30 years ago [1], and many researchers have followed this line of work [2, 3].

Chinese text consists of Chinese characters, and a number of consecutive characters form a word in the sense of English words. For instance, XUN (凶) and QI (器) are two Chinese characters, and XUN-QI (凶器) is a Chinese word approximately corresponding to *weapons* in English. SH-YONG-XUN-QI (使用凶器) is a Chinese phrase that contains two words, where SH-YONG (使用) means *use* in English, and SH-YONG-XUN-QI means *use weapons* in English.

Partially due to our ignorance, we have not been able to identify sufficient work that is directly related to indexing Chinese documents with phrases. More commonly, people segment Chinese text with the help of a machine readable lexicon, and then index the documents with Chinese words. With special techniques for obtaining information about Chinese words such as Chien’s PAT tree-based approach [4], one may segment Chinese text without using lexicons. Instead of going through Chinese word segmentation first, some have used character level bigrams for indexing Chinese documents [5, 6]. This approach offers a much improved performance than character-level indexing for Chinese text, while requiring a much larger space of index terms [7]. To further improve the quality of search results, some consider short Chinese words for indexing [7].

As an exploration toward phrase-based indexing of Chinese text, we consider word-level bigrams for indexing indictment documents in Chinese. To this end, we rely on both HowNet [8] and Chien’s PAT tree-based methods for identifying useful Chinese words. After obtaining definitions of Chinese words, we segment each document for obtaining pairs of words, and use them as the signatures of the documents. We define the similarity between indictment documents based on the number of common term pairs. Having built this infrastructure, we classify indictment documents based on their prosecution categories as Liu did in [9], and classify indictment documents based on their cited articles as Liu did in [10]. Current experimental results indicate that using term pairs leads to classification of higher quality for the former task. However, the new method provides only comparable performance on the latter task. Our methods differ from Liu’s methods for the second task in two important ways. In addition to using different indexing units, i.e., single terms vs. term pairs, we use different ways to obtain weights for these indexing units. We are still looking into the second task for further improvements.

Section 2 provides more background information regarding our work. Section 3 discusses our methods of obtaining Chinese words for the legal domain from our corpus. Section 4 extends the discussion for how we obtain phrases, how we assign weights to the phrases, and how we use the phrases for comparing the similarity between indictment documents. Section 5 contains the experimental results, and Section 6 wraps up this paper with some discussions.

2 Background

We provide more background information on details of our classification tasks. We exclude information how we segment Chinese character strings into word strings [9] for page limits. We follow a standard procedure for segmenting Chinese, i.e., preferring the longer matches while using a lexicon to determine the word boundaries, that has been adopted in the literature.

We can classify indictment documents in two different levels of grain sizes. The coarser lever is based on the prosecution categories, and the more detailed level is based on the cited articles. We consider six different prosecution categories: larceny (竊盜), robbery (搶奪), robbery by threatening or disabling the victims (強盜), re-

ceiving stolen property (贓物), causing bodily harm (傷害), and intimidation (恐嚇). We use $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_6$, respectively, to denote these categories henceforth. The criminal law in Taiwan dedicates one chapter to each of these prosecution reasons, except that the two types of robberies \mathbf{X}_2 and \mathbf{X}_3 occupy the same category.

Once judges determine the prosecution categories of the defendant, they have to decide what articles are applicable to the defendants. Each chapter for a prosecution category contains a few articles that describe applicability and corresponding sentence of the article. Not all prosecution categories require detailed articles that sub-categorize cases belonging to the prosecution category, but some prosecution categories require more detailed specifications of the prosecutable behaviors than others. In this paper, we concern ourselves with articles 266, 267, and 268 for gambling (賭博). Article 266 describes ordinary gambling cases, article 267 describes cases that involve people who make a living by gambling, and article 268 describes cases that involve people who provide locations or gather gamblers for making profits.

Applicability of these articles to a particular case depends on details of the facts cited in the indictment documents. Very simple cases violate only one of these articles, while more complex ones call for the applications of more articles. In addition, some combined applications of these articles are more normal than others in practice. Let A, B, and C denote types of cases that articles 266, 267, and 268 are applied, respectively, and a group of concatenated letters denotes a type of cases that articles denoted by each letter are applied. Based on our gathering of the published judicial documents, we observe some common types: A, C, AB, and AC. The cases of other combinations are so rare that we cannot reasonably apply and test our learning methods at this moment. Hence we will ignore those rare combinations in this paper.

Classifying indictment documents based on the cited articles is more useful than classifying documents based on the prosecution reasons, because both legal practitioners and ordinary people benefit from more exact classification. However, classifying documents based on cited articles is distinctly more difficult than classifying documents based on prosecution reasons. Documents of lawsuits that belong to the same prosecution category contain similar descriptions of the criminal actions, and sub-categorizing them requires professional training even for human experts.

3 Lexical acquisition

Although employing a machine-readable lexicon is essential for our work, relying completely on HowNet will not provide satisfactory results. HowNet was developed by excellent researchers in China, and it is not deniable that HowNet provides invaluable information about Chinese words. Nevertheless, it is also true that the Chinese languages used in Taiwan and in China have become a bit different due to the separation in the past half century. In addition, HowNet may not include all legal terms that we need. For these reasons, we employ HowNet to find useful words for the legal applications, and Figure 1 shows the flow of how we acquire the lexical information.

We apply Chien's PAT tree-based algorithm [4] and our own algorithm, TermSpotter, for spotting possible Chinese words from a training corpus. When using

the PAT tree-based algorithm, we extract only words that consist of two or three charac-

ters. We manually filter the candidate words reported by these algorithms, keeping all spotted words that were already listed in HowNet and useful words even if they are not listed in HowNet. The words are then manually clustered into categories based on their semantic similarity.

Procedure: TermSpotter (input: a training corpus; output: a list of candidate words)

1. Scan the corpus, and obtain frequencies of all bigrams
2. Concatenate bigrams that have similar occurrence frequencies into longer words, preferring those have higher frequencies
3. Save all n-grams that exceed the threshold for occurrence frequency into a wordlist
4. Remove selected words from the wordlist, and return the resulting wordlist

Our method for spotting terms in our training data is actually very simple. The TermSpotter aggregates consecutive n-grams that have similar and high occurrence frequencies into a longer word. At step 2, two neighbor n-grams will be aggregated if their frequencies did not differ more than 50% of their individual frequencies. At step 3, n-grams are considered frequent if they occurred more than 30 times. The choices of 50% and 30 were arbitrary, which make TermSpotter perform satisfactorily so far. Step 4 removes words that meet specific conditions, and we subjectively set up the conditions.

We employed both TermSpotter and Chien's algorithm to look for useful terms from 10372 real world indictment documents. The very first step in processing the legal documents that were published as HTML files was to extract the relevant sections at the preprocessing step. We then ran TermSpotter and Chien's algorithm over the corpus to get the candidate words, and manually filtered the list to obtain the keyword database. At the manual filtering step, all extracted words that were also included in HowNet would be saved in the keyword database. We subjectively decided whether to save the extracted words that were not included in HowNet. After checking each of the algorithmically extracted words, we found 1847 useful words that were already included in HowNet. We also found 832 useful words for the legal application, but they were not included in the original HowNet. In total, we have 2679 words in the keyword database.

To enhance the information encoded by the words, we manually categorized words which have similar meanings in legal applications. For instance, we have a category for *location* which includes Chinese words for *banks*, *post offices*, *night markets*, etc., and we also have a category for *vehicle* which includes such Chinese words as *passenger cars*, *busses*, *taxis*, and *trucks*. We have 143 categories that include more than two Chinese words, and we treat a word as a single-word category if that word carries a unique meaning. In categorizing the words, we ignored the problem of ambiguous words, so a word was assumed to belong to only one category. Although making such

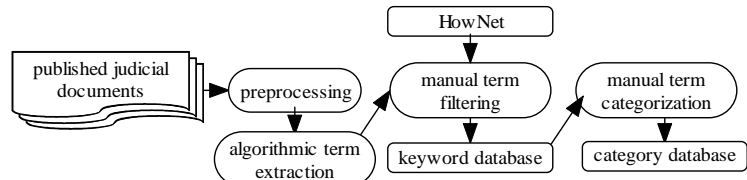


Figure 1. Constructing databases of lexical information

a strong assumption is subject to criticism, we consider it worthy of exploration because words might carry specific meanings in phrases in legal documents.

4 Phrase-based k NN classification

Instance-based learning [11] is a technique that relies on past recorded experience to classify future problem instances, and k NN methods are very common among different incarnations of the concept of instance-based learning. By defining a distance measure between the past experience and the future problem instance, a system selects k past experiences that are most similar to the future problem instance, and classifies the future instance based on the classes of the selected k past experiences.

The appropriateness of the similarity measure is crucial to the success of a k NN-based system. Given a segmented Chinese text as we explained in Section 2, we can treat each past experience as a vector or a bag of Chinese words. The distance measure can be defined in appropriate ways [12]. In this paper, we report our experience in using phrases as units of indexing for legal documents in Chinese, and in learning the weights for phrases in classifying documents. Figure 2 shows the flow for training and testing our classifiers, and major components are explained in this section.

4.1 Identifying phrases and learning their weights

Although it is intuitive that using phrases will lead to more precise indexing of the past documents, it is far less clear about how we choose phrases from sentences [1, 2, 3]. Moreover, it is not even clear that how we define “sentences” in Chinese. Although modern Chinese writing adopts punctuations such as commas and periods, a sequence of words ended with periods do not necessarily correspond to just one sentence as they normally do in English. It is very common that a sequence of Chinese words ended with a period can be translated into multiple English sentences. In this work, we choose to use commas and periods as terminators for Chinese sentences.

Given a Chinese sentence, we can segment the sequence of characters into a sequence of words. Now, with this sequence of words, can we determine a phrase that actually stands for the main idea of the original sentence? It seems that this is a tough question that we cannot solve without resorting to semantic analysis of the original text. If we may solve this problem now, we might have found a solution to the problem of exact indexing of documents which is an essential challenge for information retrieval. To circumvent this difficulty, we record all possible word pairs formed by

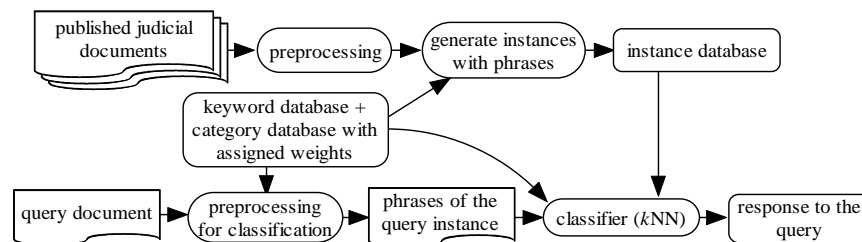


Figure 2. Training and testing our classifier

words in the sentence, and disregard those pairs which do not occur more than 10 times in the training documents. We then take the union of word pairs for all sentences in a preprocessed document as the feature list of the document. We employ this procedure in ovals labeled with *generate instances with phrases* and *preprocessing for classification* in Figure 2.

Before we explain ways of assigning weights to phrases, we elaborate on how we define phrases in more details. Assume that, after being segmented, a sentence includes three words, α , β , γ . We will preserve the original ordering of these words, and come up with three combinations, i.e., $\alpha\text{-}\beta$, $\alpha\text{-}\gamma$, and $\beta\text{-}\gamma$, as the phrases for the sentence. As a result, if we have a document that includes this sentence, these word pairs will *all* be included in the instance that represents the original document. We recognize that this might not be a good design decision, but doing so relieves us of the task of determining which phrase is the “most representative” of the original sentence for the current exploration.

It is expected that with appropriate weights, weighted k NN methods provide better performance than plain k NN methods [11]. Hence we would also like to assign weights to phrases. The weights for phrases should reflect their potential for helping us to correctly classify documents, so defining weights based on the concept similar to the inverse document frequency [12] is desirable. As we mentioned in Section 3, we actually have converted some words to their semantic categories. Hence, we will assign weights to phrases at the level of semantic category, rather than to the phrases at the word level.

We explore two methods for assigning weights to phrases. Let $S=\{s_1, \dots, s_i, \dots, s_n\}$ be the set of different types of documents in an application. Assume that a phrase κ appears f_i times in documents of type s_i . Let p_i be the conditional probability of the current document belonging to s_i , given the occurrence of κ . We may assign the quantity defined in (1) as the weight of κ . Notice that the denominator in (1) assimilates the formula of entropy. Hence a phrase with larger w_1 will collocate with fewer types of documents. We also explore the applicability of (2). Qualitatively, w_2 is similar to w_1 in that a phrase with larger w_2 will collocate with fewer types of documents.

$$w_1(\kappa) = \frac{1}{-\sum_{t=1}^n p_t \log p_t}, \quad \text{where } p_i = \frac{f_i}{\sum_{r=1}^n f_r} \quad (1)$$

$$w_2(\kappa) = \left(\sum_{t=1}^n p_t^2 \right)^2, \quad \text{where } p_i = \frac{f_i}{\sum_{r=1}^n f_r} \quad (2)$$

4.2 Similarity measure

Now that we have converted the original documents into instances that are represented by sets of phrases and that we have assigned weights to phrases, we are ready to define the similarity measure between instances for our classifier that adopts the k NN approach. Assume that we have two instances i_1 and i_2 , each representing a set of key phrases. Let $u_{1,2}$ denote the intersection of i_1 and i_2 . We explore two methods, shown in (3) and (4), for computing the similarity between i_1 and i_2 . The basic element in both (3) and (4) is the portion of common phrases in the phrases of the in-

stances being compared. Two instances are relatively more similar if they share more common phrases. Formulas (3) and (4) differ only in how we combine the two ratios.

$$s_1(i_1, i_2) = \left(\frac{\text{total weights of } u_{1,2}}{\text{total weights of } i_1} + \frac{\text{total weights of } u_{1,2}}{\text{total weights of } i_2} \right) / 2 \quad (3)$$

$$s_2(i_1, i_2) = \frac{\text{total weights of } u_{1,2}}{\sqrt{(\text{total weights of } i_1) \times (\text{total weights of } i_2)}} \quad (4)$$

4.3 More design factors

In addition to how we obtain basic words, how we define weights, and how we define similarity measure between instances, there are other design decisions that we can manipulate. It is interesting to consider whether we should take into account the part of speech (POS) of the words when we construct phrases. Since our phrases consist of only two words, it is natural for us to consider only verbs and nouns in forming the phrases. Under this constraint, we could have phrases of the form verb-verb, verb-noun, noun-verb, and noun-noun. If we interpret our phrases as basic events in the descriptions of the criminal violations, we might prefer to employ phrases led only by verbs in computing similarity between instances. Hence, we can compare the performance of classification when we consider any word pairs and when we consider only phrases with leading verbs.

The other design factor that we have considered is to limit the source of phrases. Recall the way we construct phrases in Section 4.1. If a sentence contains many basic words, we would create many combinations of these basic words, potentially introducing noisy phrases into our databases. Though the noise thus resulted may not interfere the classification very much, it may degrade the computational efficiency of the classifier. This observation leads us to screen sentences from where we would obtain phrases. For the experiment results reported in the following section, we consider sentences that contain no more than 16 Chinese characters which can be segmented into no more than 3 words. Experimental results under other settings are included in a longer version of this paper.

5 Experimental evaluations

We evaluated our classifier with real world judicial documents. Table 1 shows the quantities of the documents used for the task of classifying documents based on the prosecution categories. Table 2 shows quantities of documents for the task of classifying documents based on the cited arti-

Table 1. Number of cases in different prosecution categories

	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆
training	1600	1600	1600	1600	710	241
test	400	400	400	400	179	60

Table 2. Number of cases in different combinations of cited articles for gambling cases

	A	C	AB	AC
training	1066	802	809	344
test	267	201	203	86

Table 3. Classification based on prosecution categories (3,16)

POS	any phrases (1504)				phrases led by verbs (989)			
Weights	(1)		(2)		(1)		(2)	
Similarity	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)
<i>precision</i>	74.7%	81.9%	79.7%	85.7%	72.2%	77.1%	77.9%	85.5%
<i>Recall</i>	74.3%	67.3%	79.1%	81.5%	70.3%	63.4%	76.2%	81.5%
<i>F</i>	70.0%	68.8%	77.7%	82.1%	66.1%	63.9%	74.9%	82.2%
<i>accuracy</i>	74.9%	73.2%	81.6%	86.2%	70.7%	69.0%	78.6%	85.4%

cles. We acquired the documents from the web site of the Judicial Yuan, Taiwan (www.judicial.gov.tw). We continue to use the notation for representing different types of documents, which are discussed in Section 2.

Since our work is not different from traditional research in text classification, we embraced such standard measures as *precision*, *recall*, the *F* measure, and *accuracy* for evaluation [12]. Let p_i and r_i be the *precision* and *recall* of an experiment, *F* is defined as $(2 \times p_i \times r_i) / (p_i + r_i)$. Due to page limits, we must summarize the classification quality for all different types of documents, and we took the arithmetic average of the *precision*, *recall*, *F*, and *accuracy* of all experiments under consideration.

Tables 3 and 4 show statistics about the performance of our classifier. These tables employ the same format. The top row indicates whether we considered all types of word pairs or only phrases that were led by verbs, as we discussed in Section 4.3. The second row indicates whether we employed formula (1) or (2) for defining weights of phrases, and the third row indicates whether we computed similarity between instances by formula (3) or (4). The numbers in the top row indicate the quantities of phrases that were obtained from the training documents.

Table 3 shows the statistics for the experiments for prosecution category-based classification, when we extracted phrases from sentences which contained no more than 16 Chinese characters which were segmented into no more than three words. Using this setup, we obtained 1504 phrases when we considered all types of phrases, and, if we ignored phrases that were not led by verbs, we obtained 989 phrases from the training documents. We observed that no matter whether we consider POS of constituents of the phrases, the combination of formulas (2) and (4) would offer the best performance. This proposition held when we repeated the same experiment procedure for setups where we obtained phrases from sentences of different number of characters and words. Results for classifying cases based on cited articles, i.e., statistics in Table 4, further support that (2) and (4) together outperform for the task of cited article-based classification than other combinations of the formulas. Assuming that we use (1) for defining weights, (3) seems to be a better choice for

Table 4. Classification based on cited articles (3,16)

POS	any phrases (459)				phrases led by verbs (262)			
Weights	(1)		(2)		(1)		(2)	
Similarity	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)
<i>precision</i>	77.8%	73.9%	79.5%	80.6%	75.2%	74.7%	77.6%	77.9%
<i>Recall</i>	79.0%	75.4%	80.5%	81.7%	76.0%	76.1%	78.7%	79.0%
<i>F</i>	77.4%	73.8%	78.8%	80.2%	75.0%	74.8%	77.3%	77.7%
<i>accuracy</i>	78.9%	75.4%	80.8%	81.9%	77.0%	76.1%	79.3%	79.7%

computing similarity between instances.

Statistics, particularly those for the combination of (2) and (4), in both Tables 3 and 4 do not show any relative superiority in classification quality for whether we should consider POS of the constituents of the phrases. We do not consider the differences in the statistics significant although the averages for not considering POSs seem a bit better. However, we would have used more than 40% of number of phrases in the classification for considering all types of phrases. Using phrases that were led by verbs clearly had an edge on computational efficiency.

Corresponding numbers for the combination of (2) and (4) in Tables 3 and 4 support the intuition that classification based on prosecution categories is relatively easier than classification based on cited articles. The same observation had been reported by Liu and Liao [10]. However, we have to interpret this indication of our statistics carefully, because all cases that were used for obtaining Table 4 committed the crime of gambling in different details, while cases for obtaining Table 3 belonged to a range of different prosecution categories not including gambling, as we reported in Section 2. In addition, corresponding numbers for other columns in Tables 3 and 4 do not fully support the intuition. Cases that belong to prosecution categories in our experiments may contain related criminal violations that disoriented our classifier. For instance, it should not be surprising that one would describe something that related to larceny (X_1) before one could describe how one received stolen property (X_4). Therefore differentiating X_1 and X_4 may not be easier than telling A and AC apart for gambling cases.

6 Concluding remarks

We reported an exploration among a myriad of possible ways of applying phrases to classifying judicial documents. The preliminary results are encouraging. Compared with the results reported in [9], we are able to achieve better quality of classification when our target prosecution categories are much closer than those used in previous studies. For the task of classifying cases based on cited articles, our method provides comparable quality with those reported by Liu and Liao [10]. However, our training method is more succinct and easy to understand and implement than the introspective learning method used in [10]. Our current system employs only relative frequencies of phrases among different types of documents, but Liu and Liao had to learn and adjust weights for each keyword in each training instance. Nevertheless, our advantage may have come from that we have to manually filter the Chinese words, and Liu and Liao's approach does not need human intervention at all. Similar to Liu's results, our results are better than those reported by Thompson [11]. Our results are also better than those achieved by pioneers of the phrase-based approach [2, 3], partially because the legal domains employ more specific terms in the judicial documents.

Besides the inspiring results, the exploration left us more questions to study. Indeed, pioneers had reported many challenges for the phrase-based approach [2, 3]. It should be clear that coming up with an effective method for weighting the phrases is not easy. It is also difficult to determine how we obtain important phrases for

indexing the case instances. Our two types of phrases correspond somewhat to statistical phrases and syntactic phrases [3]. Our results concur with Fagan's in that syntactic phrases do not provide significant better performance than statistical phrases. Nevertheless, we cannot be satisfied with the current results, and would like to study related issues for fully understanding the applicability of phrase-based indexing for specific domains such as legal case classification in Chinese.

Acknowledgements

This research was funded in part by contract NSC-94-2213-E-004-008 of the National Science Council of Taiwan. We thank the reviewers for their unreserved and invaluable comments. Unfortunately, we cannot add more contents for responding to the comments when we have to shorten the submitted paper for page limits already. A more complete version of this paper is available upon request.

References

ICAIL stands for *Int. Conf. on Artificial Intelligence and Law*, and *SIGIR* for *Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*.

1. G. Salton, C. S. Yang, and C. T. Yu, A theory of term importance in automatic text analysis, *J. of the American Society for Information Science*, **26**(1), 33–44, 1975.
2. J. L. Fagan, Automatic phrase indexing for document retrieval, *Proc. of the 10th SIGIR*, 91–101, 1987.
3. W. B. Croft, H. R. Turtle, and D. D. Lewis, The use of phrases and structured queries in information retrieval, *Proc. of the 14th SIGIR*, 32–45, 1991.
4. L.-F. Chien, PAT-tree-based keyword extraction for Chinese information retrieval, *Proc. of the 20th SIGIR*, 50–58, 1997.
5. I. Moulinier, H. Molina-Salgado, and P. Jackson, Thomson Legal and Regulatory at NTCIR-3: Japanese, Chinese and English retrieval experiments, *Proc. of the 3rd NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering*, 2002.
6. L.-F. Chien, Fast and quasi-natural language search for gigabytes of Chinese texts, *Proc. of the 18th SIGIR*, 112–120, 1995.
7. K. L. Kwok, Comparing representations in Chinese information retrieval, *Proc. of the 20th SIGIR*, 34–41, 1997.
8. HowNet. <www.keenage.com>
9. C.-L. Liu, C.-T. Chang, and J.-H. Ho, Case instance generation and refinement for case-based criminal summary judgments in Chinese. *J. of Information Science and Engineering*, **20**(4), 783–800, 2004.
10. C.-L. Liu and T.-M. Liao, Classifying criminal charges in Chinese for Web-based legal services, *Proc. of the 7th Asia Pacific Web Conf.*, 64–75, 2005.
11. T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
12. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.
13. P. Thompson, Automatic categorization of case law, *Proc. of the 8th ICAIL*, 70–77, 2001.

An Experience in Learning about Learning Composite Concepts

Chao-Lin Liu and Yu-Ting Wang

Department of Computer Science, National Chengchi University, Taiwan
chaolin@nccu.edu.tw

Abstract

Students need to integrate multiple basic concepts to become competent in the activities that require the knowledge of the composite concept. Traditionally, we rely on experts' judgments to build models for this integration process. In this paper, we explore computational methods for unveiling how students learn composite concepts, and compare effects of applying mutual information-based and hierarchical search-based techniques for guessing the unobservable processes, which were simulated by Bayesian networks. Experimental results show that computational methods can be useful in assisting this student modelling task.

1. Introduction

We continue our exploration of student modeling with Bayesian networks [2]. In particular, we try computational methods for learning how students learn composite concepts. Our learning component takes as input the item response patterns (IRPs) of simulated students, and returns the best Bayesian network that may explain the behavior of the simulated students.

Using Bayesian networks (BNs) to model students' knowledge structure is not a brand new idea. Millán et al. proposed four-level networks which included nodes for *subjects*, *topics*, *concepts*, and *questions* [4]. They offered evidence on how BNs of higher quality improved the efficiency of computerized-adaptive testing [1]. Although the directions of arcs in BNs do not necessarily correspond to causal directions in educational applications [3,4], and we are not discussing this issue in this issue, we follow the most popular way to apply BNs for student modeling.

Learning the conditional probability tables, e.g., [5], and learning the structures, e.g., [6], of BNs from students' data are not new either. The latter is relatively rarely discussed in the literature, however, partially because most practical systems employ experts' opinion in choosing the network structures. We explore the issue of whether computational techniques can help us choose the best model from a set of competing models. Our approach is different with Vomlel's [6] in how we utilized the experts' opinion. We simulate students'

behavior based on the competing structures, and apply the simulated data to guess students' learning process. We hope the results of this study can shed light on how we can learn models of real students from real data.

In this abbreviated paper, we review how we simulated students' behavior in Section 2. We explain the basic idea of applying mutual information for comparing BNs in Section 3, and discuss a search-based method for identifying the Bayesian network that we used to generate the simulated data in Section 4.

2. Preliminaries

We created item response patterns of simulated students with our simulation environment [2]. We specified the network structure, the competence patterns, and the range of uncertain relationships between students' competence levels with their item responses in a simulation. Figure 1 shows a network structure, where we have seven concepts, three test items designed for each concept, and one node named 'group' for encoding students of different competence patterns. Nodes whose names begin with *c* and *d* denote basic and composite concepts, respectively. In this network, the modeler encodes the belief that students learn *dABC* by directly integrating the three basic concepts, which is indicated by the direct links between the nodes.

The competence patterns assign stereotypical behavior of different types of students, and need to be given in a matrix form (the Q matrix in [2]). By varying the contents of the matrices, we simulated different competence combinations of the student population.

The simulator considers typical uncertain relationships between students' item responses and their competence levels, i.e., *guess* and *slip* [3,4]. This is controlled by a simulation parameter called *fuzziness*. Stu-

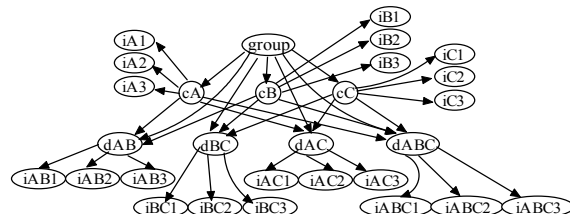


Figure 1. A BN for 3 basic concepts [2,3]

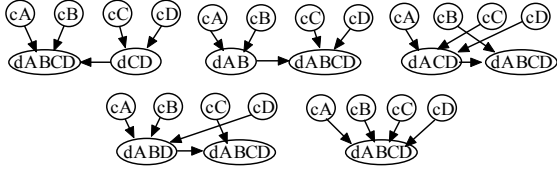


Figure 2. Candidate BNs in our experiments

dents may also behave differently from their typical competence patterns, and the degree of such abnormal behavior is controlled by *groupInfluence*. Using larger values for these parameters will introduce more randomness into the observed item response patterns.

Due to the size of the compete network, we do not show the networks that we used in our experiments completely. Figure 2 shows five substructures of the BNs that we used to simulate students' item responses. We considered problems involving four basic concepts, and we would like our programs to learn how students learn *dABCD*. We assume that students learn a composite concept from non-overlapping parent concepts, i.e., they do not share common basic concepts. This simplifies the problem space, but does not make the problem trivial.

3. MI-based heuristics

If we pretend that we were able to directly observe the states of the concept nodes, we can apply mutual information-based measures to our task. The mutual information between the composite concept and its parent concepts should be large than others. Let $MI(X;Y)$ denote the mutual information between two sets of random variables X and Y . If the true structure is A_B_CD (i.e., learning *dABCD* by integrating *cA*, *cB*, and *dCD*), then $MI(cA, cB, dCD; dABC)$ should be larger than $MI(dAB, cC, cD; dABC)$ and other MI measures. Analogously, if the true structure is AB_C_D , then $MI(dAB, cC, cD; dABC)$ should be the largest among all MI measures for all competing structures.

We have assumed that students will respond to three test items for each concept in Section 2, so students may correctly answer 0%, 33%, 67%, or 100% of the test items for a concept. We can use this percentage as the estimate for the state for a concept node, and, similarly, we can estimate the joint distributions of multiple concept nodes. For instance, $\Pr(dab=33\%, cc=67\%)$ was set to the percentage of students who correctly answered one item and two items, respectively, for *dAB* and *cC*. We also have to smooth the probability distributions to avoid zero probabilities because some configurations of the involved variables may not appear in the simulated samples. We add 0.001 to the number of occurrence of every different configuration of the variables. With this procedure, we have a way to estimate the mutual information measures, and we can try the following

ures, and we can try the following heuristics in experiments.

Heuristics: The competing structure that has the largest mutual information measure is the hidden structure.

4. Search-based model selection

Instead of computing the MI measures for all competing structures, it is possible to do the comparison incrementally, and we have a search-based procedure.

We illustrate the search procedure in Figure 3. The filled circle represents the beginning of the search procedure, and the search goes from the left to the right. We compute the estimated MI (EMI) of the competing structures in which *dABCD* has only two parent concepts. The structure that has the largest EMI becomes the current candidate. We then compute the EMIs of the successors of the candidate. In Figure 3, structures on the second to the leftmost column are connected to their successors on the second to the rightmost column by

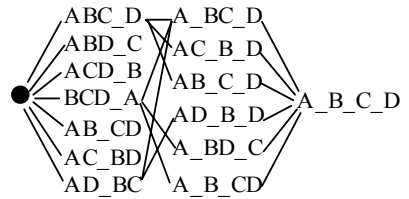


Figure 3. The search space (partially shown for readability)

lines. *Successors* are structures that are refined from the original candidate to include exactly one more component than the original candidate. We do not show all the lines in the middle of the graph for readability. If the largest EMI of the successors is smaller than the EMI of the current candidate, then the current candidate is the answer. Otherwise, the successor that has that largest EMI becomes the current candidate. In the latter case, we will have to compute the EMI of $A_B_C_D$, which must be a successor of the new candidate in Figure 3. If the EMI of $A_B_C_D$ is larger than that of the new candidate, then $A_B_C_D$ is the answer, otherwise the new candidate is the answer.

This search procedure is recursive, and can be expanded and applied to more complex situations when there are more than four basic concepts.

In the experiments, we set *fuzziness* and *groupInfluence* to different combinations of 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3. We did not try values larger than 0.3 because values larger than 0.3 were not considered in the literature, to the best of our knowledge. Hence, we conducted 36 experiments. In each of these 36 experiments, we created 600 different networks for each of the five competing structures in Figure 2. Each of these networks was given different underlying joint probability distributions. In order to compute reliable mutual information, we randomly sampled the IRPs of 10000 simulated students from each network.

The chart shown in Figure 4 shows a comparison between the effects of using the heuristics, discussed in Section 3, and the search-based method. The horizontal axis shows the decimal parts of the values of *fuzziness*. The legend shows where the data for the curves came from and the decimal parts of *groupInfluence*. For instance, ‘s05’ indicates that the search method was used when *groupInfluence* was 0.05. The vertical axis shows the percentage of correct prediction of the hidden structures of the 3000 (=5×600) different networks in an experiment.

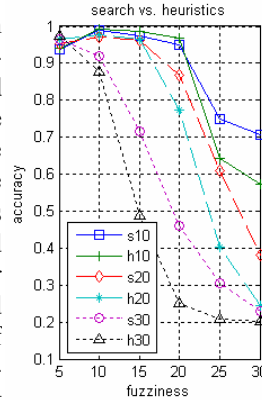


Figure 4. Search outperforms pure heuristics

Beyond this range, both methods did not perform very well, but the search-based method offered similar or better predication than using only the heuristics.

In the most challenging case when both *groupInfluence* and *fuzziness* were set to 0.30, the accuracy for the heuristics-based method was only 20%, which is equal to what one would get for a random guess among five alternatives. This is an interesting observation, because we allowed our classifier to guess any of the fourteen possible ways shown in Figure 3, and a random guess should have given about 7%. This phenomenon is related to two factors: that we used basic concepts as the parent concepts of all composite concepts, except *dABCD*; and that the basic concepts must be ancestors of *dABCD*, although they might not be the parent concepts of *dABCD*. As a consequence, computing the EMIs as we defined in Section 3 offered a special favor to the structure *A_B_C_D*, and the accuracy happened to be equal to the results of random guess among the five true answers shown in Figure 2 where the possible answers included *A_B_C_D*. If we had excluded the *A_B_C_D* cases from the test data, the accuracy would fall below 25%, which is the result of random guesses if there were four possible answers.

It is also interesting to find that, when the degree of *fuzziness* reduced, the accuracy did not improve all the time (the upper left corner of the charts). After examining the confusion matrices, we found that our programs misclassified many *A_B_CD* and *AB_C_D* structures as *AB_CD*. Intuitively, this type of error is understandable because *AB_CD* is really close to the true

answers. The percentage of this type of errors is related to the settings for other composite concepts in the Q matrix, which we will explain in an expanded version of this manuscript.

5. Concluding remarks

Learning how students learn composite concepts is an interesting and challenging task. We need to infer about the internal process from students’ external behaviours that have only indirect and probabilistic relationship with the internal states. Experimental results showed that it is possible to guess the correct answer with the search-based method if the degree of uncertainty is limited. When the degree of uncertainty is moderately large, it is better to consult experts, apply experts’ knowledge to train classifiers that employ artificial neural networks (ANNs) or support vector machines, and use the classifiers to guess the hidden structure. Figure 5 shows a snapshot of the results of applying ANNs for this task that we will report in an extended report.

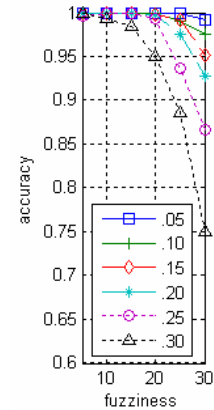


Figure 5. Using ANN classifiers

Acknowledgements

We thank reviewers for their invaluable comments on a manuscript of this paper. We take the comments seriously, though this shortened paper definitely cannot reflect our appreciation. This research was supported in part by the research contract 94-2213-E-004-008 of National Science Council of Taiwan.

References

1. C. Carmona, E. Millán, J. L. Pérez-de-la-Cruz, M. Trella, and R. Conejo, Introducing prerequisite relations in a multi-layered Bayesian student model. *Lecture Notes in Computer Science* 3538, 347-356, 2005.
2. C.-L. Liu, Using mutual information for adaptive item comparison and student assessment, *J. of Educational Technology & Society*, 8 (4), 100-119, 2005.
3. C.-L. Liu, Using Bayesian networks for student modeling, in *Cognitively Informed Systems: Utilizing Practical Approaches to Enrich Information Presentation and Transfer*, E. M. Alkhalifa (ed.), 282-309, 2006.
4. E. Millán and J. L. Pérez-de-la-Cruz, A Bayesian diagnostic algorithm for student modeling and its evaluation, *UMUAI*, 12 (2-3), 281-330, 2002.
5. R. J. Mislevy, R. G. Almond, D. Yan, and L. S. Steinberg, Bayes nets in educational assessment: Where do the numbers come from? *15th UAI*, 437-446, 1999.
6. J. Vomlel, Bayesian networks in educational testing, *IJFUKS*, 12 (Supplement 1), 83-100, 2004.

Learning How Students Learn

Chao-Lin Liu

National Chengchi University, Taipei 11605, Taiwan, chaolin@nccu.edu.tw

Abstract

This extended abstract summarizes an exploration of how computational techniques may help educational experts identify fine-grained student models. In particular, we look for methods that help us learn how students learn composite concepts. We employ Bayesian networks for the representation of student models, and cast the problem as an instance of learning the hidden substructures of Bayesian networks. The problem is challenging because we do not have direct access to students' competence in concepts, though we can observe students' responses to test items that have only indirect and probabilistic relationships with the competence levels. We apply mutual information and backpropagation neural networks for this learning problem, and experimental results indicate that computational techniques can be helpful in guessing the hidden knowledge structures under some circumstances.

Summary

Behavior models of activity participants are crucial to the success of computer systems that interact with human users. When using Bayesian networks (BNs) as the language for model construction, Mislevy et al. asked where we could obtain the numbers for the conditional probability tables (CPTs) [1]. We could ponder where we could obtain the structures of the BNs in the first place. For educational practitioners, an obvious and practical answer to this inquisitiveness may be that we should consult experts of the targeted domains to provide the knowledge structures, such as the prerequisite relationships between concepts, for building student and instructor models. Indeed this is an effective and the de facto approach to building computer-assisted educational software in general. Can computers be more helpful than finding the detailed numbers in the CPTs for student modeling? More specifically, can computers assist in any way for finding the structures of student models? Given a *composite concept*, say $dABC$, that requires knowledge about three *basic concepts*, say cA , cB , and cC , how can we tell how students learn $dABC$ from cA , cB , and cC ? Do students combine cA and cB into an intermediate product, dAB , and then combine dAB and cC into $dABC$? Or, do students integrate the basic concepts directly to learn $dABC$?

In this exploration, we assume that students learn the composite concept from ingredient constructs that do not include overlapping basic concepts. For instance, we subjectively exclude the possibility of learning $dABC$ from two

intermediate composite concepts dAB and dBC , because they both include cB . This assumption simplifies the search space. However, the size of the search space still grows explosively with the number of basic concepts included in the target composite concept, and is related to the Stirling number of the second kind.

We assume that educational experts provide a set of possible ways that students may, implicitly or explicitly, employ to learn the composite concept, and our job is to help experts identify which of these learning patterns is the most likely answer. Hence, the process of learning how students learn begins with the acquisition of a set of candidate answers. We use the set of candidate learning patterns to build BNs for simulating possible student behaviors, and employ the simulated data to train backpropagation neural networks (BPNs). The learned BPNs can then be used to classify the unobservable learning pattern, based on students' item responses, into one of the candidate answers.

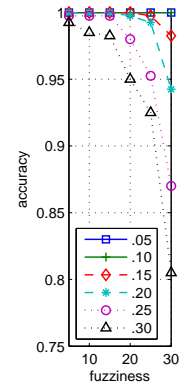
Following the steps of many researchers who explored methodologies for building computer-assisted tutoring systems, we employ simulated students in this study. Simulated students were generated from Liu's simulation system that considers the probabilistic relationships between students' responses to test items and students' competence levels in concepts [2]. The degree of uncertain relationship between these two factors was controlled by a parameter called *fuzziness*. We set *fuzziness* to a larger value when we simulated a more uncertain relationship between responses to items and competence in concepts. The other parameter, named *groupInfluence*, affected the uncertain relationship between the students' actual behaviors and students' stereotypical behaviors. We set *groupInfluence* to a larger value to make students more likely to deviate from their typical behaviors. In short, it became harder to guess the real mental states of a student when either *fuzziness* or *groupInfluence* were set to larger values in the simulation.

Students' responses to test items and students' competence levels were represented with different, though directly connected, nodes in the BNs that were used to generate simulated students. States of nodes that represented competence levels in concepts were not observable, and only states of nodes that represented correctness of item responses were accessible. Hence, our job was to guess the substructure of the unobservable nodes based on the data that had only indirect and probabilistic relationships with the true answers. Due to this reason, known algorithms for learning structures of Bayesian networks, such as the PC algorithm implemented in Hugin, were not directly viable for this learning problem.

We employed estimated mutual information (EMI) for comparing the candidate solutions. If students learn $dABC$ from dAB and cC rather than from cA and dBC , the EMI between the nodes for both dAB and cC and the node for $dABC$ may be larger than the EMI between the nodes for both cA and dBC and the node for $dABC$. (In this case, $EMI(dAB, cC|dABC)$ is expected to be larger than $EMI(cA, dBC|dABC)$.) Namely, we used the EMI to represent the merits of a competing substructure. We had to estimate the mutual information between two sets of nodes, since we did not have direct access to the states of the nodes that represented concepts. We estimated the state for the node that represented a concept with the percentage of correct responses to test items designed for the concept, and used the estimated states of nodes to calculate the

EMIs. In addition to the EMIs for all competing substructures, we introduced ratios between the EMIs for training the BPNs. Experience indicated that ratios between the EMIs, e.g., the ratios between the EMIs and the largest EMI, were useful for improving the prediction quality of the trained BPNs.

We tested the proposed procedure for guessing how students learn $dABC$. There were four possible answers. We randomly sampled 500 network instances that had different underlying joint probability distributions for each of these four answers, and simulated item responses of 10000 students that were generated from these 2000(=4×500) networks. Each simulated student responded to three items for seven concepts, i.e., cA , cB , cC , dAB , dAC , dBC , $dABC$, and the responses must be either correct or incorrect. We calculated the EMIs and their ratios for each network instance for training BPNs, so we trained the BPNs with 2000 training instances. We then applied the trained BPNs to predict the learning patterns of 400 groups of students—100 groups generated for each of the four answers. We repeated the above procedure for 36 combinations of *fuzziness* and *groupInfluence*, each ranging between 0.05 to 0.30. The figure on this page shows the results. The horizontal axis shows the decimal part of *fuzziness*, the legend shows the values of *groupInfluence*, and the vertical axis shows the percentage of correct identification of hidden structures in 400 test cases. The results suggest that it is possible to identify the hidden structure better than 80 percent of the time, if *fuzziness* and *groupInfluence* are not large and if educational experts' guess list does include the correct structure.



Do we really need student models of better quality? Experimental results reported by Carmona et al. suggested that student models of higher quality could help us improve the effectiveness of computerized adaptive tests [3]. Hence, we hope results outlined in this extended abstract can be useful. We have expanded our experiments to cases where we learned how students learn composite concepts that included four basic concepts [4]. The accuracy remained above 75% in unfavorable conditions. We thank reviewers for their invaluable comments on the original manuscript. This work was partially supported by the research contract 94-2213-E-004-008 of National Science Council of Taiwan.

References

1. Mislevy, R.J., Almond, R.G., Yan, D., Steinberg, L.S.: Bayes nets in educational assessment: Where do the numbers come from? 15th UAI (1999) 437–446
2. Liu, C.L.: Using mutual information for adaptive item comparison and student assessment. *J. of Educational Technology & Society* **8**(4) (2005) 100–119
3. Carmona, C., Millán, E., Pérez-de-la-Cruz, J.L., Trella, M., Conejo, R.: Introducing prerequisite relations in a multi-layered Bayesian student model. *Lecture Notes in Computer Science* 3538 (2005) 347–356
4. Liu, C.L., Wang, Y.T.: An experience in learning about learning composite concepts. 6th IEEE ICALT (2006) to appear

行政院國家科學委員會補助專題研究計畫 出國報告

分類技術與貝氏網路之應用： 法學文件之語意標記與人機互動之使用

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 98-2213-E-004-008

執行期間： 94 年 8 月 1 日至 95 年 10 月 31 日

計畫主持人：劉昭麟

共同主持人：

計畫參與人員：黃珮雯、鄭人豪、陳禹勳及林仁祥

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立政治大學 資訊科學系

中 華 民 國 95 年 10 月 4 日

Report for Attending ISMIS 2006

Chao-Lin Liu

Department of Computer Science, National Chengchi University

ISMIS 2006 was held in Bari, Italy between 27 and 29 September 2006. I presented two papers during this conference on student modeling and legal informatics, and this is a brief report for the trip to ISMIS 2006.

About ISMIS

The International Symposium on Methodologies for Intelligent Systems (ISMIS) has a long history, and this sixteenth ISMIS was held in Bari Italy between 27 and 29 September 2006. Normally, there are two ISMISes for every three years. The main initiator and organizer of this series of conferences is Professor Ras of the University of North Carolina at Charlotte. The next ISMIS will take place in Toronto Canada in May of 2008.

According to the report provided by the program chair of ISMIS 2006, 192 papers were submitted from 34 countries. Among these 192 submitted papers, 66 papers were accepted as long papers, and 15 papers were accepted as short papers. The acceptance rate for long papers was about 34%. Including the short papers, the overall acceptance rate for ISMIS 2006 reached 50%.

Taiwan, followed immediately by Japan, ranked sixth in terms of submitted papers. The leading countries are China, Italy, South Korea, USA, and France. Surprisingly the acceptance rate achieved by the Chinese papers was just about 10%, suggesting that there were just about 5 papers authored by people with Chinese identity. The acceptance rate for Taiwanese papers was about 70%, meaning that five papers were accepted. If you have access to the conference proceedings, you will find that they consist of four long papers and one short paper. In addition, these five Taiwanese papers were authored by three groups. NCCU had two long papers, KAUS had one long and one short papers, and TKU had one long paper. Unfortunately, NCCU is the only group that did show up and present the papers.

There was a lady who actually came from the Hampton University in Virginia USA, but she went to USA from Taiwan. Li-Shiang Tsay is a fresh professor at Hampton and a former student of Professor Ras of the UNC.

About Chinese presence

Due to the relationship between Taiwan and China, I thought it is appropriate to say something about Chinese presence in this report.

During this conference, I did not have chances to contact with any Chinese who came directly from China. However, I did have chances to meet people who left China for other countries. I met two from Japan, two from USA, and one from German.

If we boldly take the status of attendance as a sign, Taiwanese presence is significantly lower than Chinese presence. Chinese people have worked very hard in the past years to increase their international presence, supported by their huge population and strengthening economic situations. The status quo at ISMIS 2006 can be just one of the signs.

About the quality of ISMIS 2006 papers

It is almost impossible to summarize the quality of 81 papers with just a simple statement. Generally speaking, I think ISMIS could be ranked as a leader in the second tier AI conferences. The organization of the papers is generally good. Like many second tier conference, most papers discussed applications of known techniques or marginal changes to known techniques. The worst thing that I would mention is that there are people who did not show up for their presentation.

The keynote speeches were interesting. I attended one given by Steffen Staab and one given by Ivan Bartko. The main theme of Staab's talk was semantics for multimedia annotation, and the main theme of Bartko's talk was agent-based machine learning. Staab discussed current and possibly future approach to annotating multimedia material with semantic information. Bartko presented his work on using agent's low level experience in help rule learning, which sounded like explanation-based learning discussed in general AI textbooks.

The proceedings of ISMIS 2006 was published by Springer Verlag in Lecture Notes of Computer Science 4203. It should be easy to look into more details for the ISMIS 2006 presentations in digital libraries.

About English and Italian

It seems that Italy is not a country that likes to talk in English. This is definitely a contrary example against how Taiwan has tried to push her people to learn and speak English. After entering Bari and leaving the Bari airport, English became useless. Even after I checked into the Hotel Campus, I did not find any TV channels that speak English. Checking FM radios, I did not hit any luck either. I asked an American who also attended ISMIS about his hotel, located in the city, about this situation. He showed his unhappiness that he could not find English-speaking channels in a high quality hotel.

This phenomenon is interesting because we find an industrialized country that does not emphasize English. Before I left Italy, I learned few everyday words in Italy, and I think they are useful. (Note: Returning to Rome, the situation changes, and more people can speak and understand English.)

About the similarity between Italy and Taiwan

Before I went to Italy, I heard that Italian and Chinese are similar. After coming to Rome and Bari in person, I observed supporting evidence. I mention some of them here.

You can see many honking cars on Italian streets like you might many years in Taiwan. People in Italy may have walk through the traffic to cross streets. That was not always because cars do not yield to pedestrians every time, but because not all pedestrians want to wait for the green lights.

Many people I met in Bari were very kind. They tried to tell me how to go to the University of Bari, even though they could not speak English.

In addition, you may find dogs' shit on Italian streets, though I did see people clean their dogs' remains. This embarrassing problem also exists in Taiwan.

About the venue of ISMIS 2006

Bari is located on the eastern coast of Italy. It is a city for both industry and trade. Due to the importance of its location, this city was once occupied by Greece. The city also hosts the tomb of

San Nicola, who represents the origin of the Christmas. I had the chance to visit the church in the name of San Nicola, but it is forbidden to take pictures in there. The church has painting on the ceiling, and is quite a place to visit. Because of its geographic location and importance in the area, Bari also has a fort in the city. It is also recommended to visit this fort if at all possible.

About funding researchers for international presence

Once again, I would like to take this opportunity to recommend that the governments should support researchers for attending international conferences. Even if they do not publish papers, the government should also provide limited fund for them to observe international conferences. This does help internationalization and people's mutual understanding. With sufficient incentives, researchers may have a stronger motivation to work harder and publish good papers.