Topic Extraction from Item-Level Grades

Titus Winters, Christian Shelton, Tom Payne, and **Guobiao Mei**
University of California, Riverside
Riverside, California 925216
{titus,cshelton,thp,gmei}@cs.ucr.edu

Abstract

The most common form of dataset within the educational domain is likely the course gradebook. Data mining on the assignment-level scores is unlikely to provide meaningful results, but a matrix recording scores for every student and every question may provide hidden insight into the workings of a course. Here we will investigate collaborative filtering techniques applied to such data in an attempt to discover what the fundamental topics of a course are and the proficiencies of each student in those topics.

Nearly every university-level course offering creates a new education-related dataset in the process of recording student grades. While the vast majority of gradebooks record each student's aggregate score on each assignment, it is generally the case that each question was independently graded before those scores were aggregated. With suitable grading tools, these scores can be recorded at the same level of detail without significant extra work. Such a dataset contains more information than one that has aggregated out much of the meaningful information by summing up student's item-level scores.

Given item-level scores for a course, there are a number of questions that seem obvious to ask. The question "Which items are testing the same information?" could be viewed as an application of clustering. The question "How many topics are there in this course?" could be answered with hierarchical clustering techniques, or matrix decompositions (such as singular value decomposition). The question "Which are the most predictive questions in the course?" could be solved with any number of regression techniques, given a final numeric course grade. In this paper, we are concerned with the question "What are the fundamental topics that this course really tests, and how well did each student do in those topics?" or symmetrically, "What are the topics in this course and which questions test which topics?" These questions motivate the use of collaborative filtering techniques on this educational data.

Manually identifying topic information for each question can be tedious, and may not match the statistical model supported by the data itself. An ability to automatically extract topic information from student scores would avoid such tedium and allow instructors to

- determine which topic a student is struggling most with,
- give targeted study suggestions,
- build a question-bank of questions with known difficulty levels and known relevancy to each topic in the course,
- automatically generate a test or quiz covering specific topics with a known expected average score, or
- compare the data-supported topics with the instructor's view on the topics to gain a better understanding of the mental model of the students.

A simple approach to this problem would be to apply clustering algorithms to the score information, using the score for each student as an entry in a large vector and building clusters of those vectors. But to correctly answer a given question may require knowledge of multiple topics. Additionally, the amount of "missing" data due to student absences makes the use of clustering algorithms difficult. By contrast, techniques from the areas of *collaborative filtering* and *common-factor analysis* are candidates for addressing such issues.

This paper investigates the use of a number of algorithms on item-level gradebooks. Section 2 describes our input format and the terminology. Section 3 introduces the filtering algorithms to be investigated. Section 4 presents results for each of the algorithms and Section 5 summarizes our contributions.

It is important to note that the concepts here can be reproduced to a degree using well-known techniques from educational statistics. The most similar technique from that field is common-factor analysis, which we will compare against directly. Using Cronbach's alpha statistic from classical test theory allows us to measure the extent to which a set of questions is measuring the same concept. Algorithms based on this are possible, but will not allow the addition of outside (non-score) information. For the sake of this extra flexibility, we focus here primarily on data-mining techniques.

2. Terminology and Notation

Following the terminology of educational statistics, the term *item* will represent anything for which per-student grades are recorded: questions, problems, parts of questions, etc. Also,

rather than saying "assignment," "test," or "quiz," we will use the term *instrument*.

The data we will be working with can be represented as an $m \times n$ matrix M where m is the number of students in the course and n is the number of items that were assessed in the course. Some algorithms may make use of an additional m-vector S, namely the vector of overall course scores per student. Additionally we assume we have access to a listing of which items are given in the same instrument. As a preprocessing step, the entries of each column of M are scaled to be in the interval $[0, 1]$.

It is important to note that M is unlikely to be complete. In general, there will be students that do not take one or more instruments during the course offering. If these algorithms are applied to a single instrument, there should be no missing data since the rows for absent students can be ignored entirely. Processing the gradebook as a whole will reduce the number of algorithms that are applicable, but it will allow for the discovery of patterns and connections across instruments. Whole-course processing is recommended, and indeed the techniques presented here work with missing data.

3. Collaborative Filtering

Collaborative filtering was initially intended as a method for filtering arbitrary information based on user preferences (Goldberg *et al.* 1992), for use in systems such as Amazon.com's recommendation engine. At a high level, a pool of users assign ratings to a various products (books, movies, CDs) and the system infers the underlying structure: what factors govern whether a user will like a product or not, how much does each factor affect a given user, and how much of each factor is present in each product. User preferences may drift over time, but most filtering algorithms ignore this.

The overall concept of collaborative filtering can be viewed in a number of ways. Collaborative filtering can be viewed as a system for predicting missing ratings based on a user's similarity to other users that rated the missing item. It can also be viewed as identifying the latent factors that influence each rating and estimating the influence of those factors on each user and each product.

This concept generalizes very easily into the educational domain, although the input matrices are generally denser (fewer missing values) in educational datasets. Each course covers a number of "topics," which are the latent factors in this context. Each item in a course pertains to one or more of these topics, and each student has some ability in each topic. Students may gain or lose ability in each topic over the course of the quarter, just as user preferences may drift. Here we assume that drift is a negligible effect since testing on a topic generally occurs *after* that topic has been covered in the course.

Non-Negative Matrix Factorization

A well known approach to collaborative filtering is Non-Negative Matrix Factorization(Lee & Seung 2001) (NMF). Under NMF, the matrix of interest M is approximated by the product of two non-negative matrices $WH \approx M$ where W is $m \times f$ and H is $f \times n$. The parameter f is the number of factors assumed to be present in the data. Additionally, each entry in M, W, and H is restricted to be non-negative. In the idealized educational view of the algorithm the factors are topics, i.e., $W_{i,j}$ is the ability of student i in topic j, and $H_{j,k}$ is the relevance of topic j to question k. The matrix W is the student ability matrix, and H is the question relevance matrix. In practice the factors that are identified may correspond to non-topical factors ranging from whether the student was having a good day to how good the student is at reading trick questions.

The non-negativity of NMF gives the output of the algorithm a more intuitive interpretation than the factors from principal-component analysis (PCA) or singular-value decomposition (SVD) where the matrix is broken down into possibly negative values. With NMF, any non-zero entry is an additive weight for the final output. The assumption that a student's score on an item is the inner product of their ability on f topics and the relevancy of the item to those f topics is a simplification, but it does capture some of the structure of the process.

A straightforward technique presented in (Lee & Seung 2001) is a gradient descent approach to discovering the W and H whose product best approximates M. At each iteration, every entry in W and H is updated multiplicatively in a way guaranteed to reduce the component-wise Euclidean distance between M and WH. On most if not all gradebook datasets, this algorithm converges to within 1% of M in 250 to 500 iterations, each of which can be calculated rapidly. Additionally, it is generally very insensitive to initial conditions. Of the 1000 NMF runs we have performed to date, only once has there been a noticeable effect of the algorithm being trapped in a local minimum. The simplest algorithm presented gives multiplicative update rules that perform a gradient descent minimization of the Frobenius norm of $M - WH$ (that is, the element-wise Euclidean distance between M and WH):

$$H_{a,u} \leftarrow H_{a,u} \frac{[W^T M]_{a,u}}{[W^T W H]_{a,u}}$$

$$W_{i,a} \leftarrow W_{i,a} \frac{[M H^T]_{i,a}}{[W H H^T]_{i,a}}$$

However, this relies upon full data due to the matrix-multiplication steps involving M in the update rules. Since M is unlikely to be complete, this version of the NMF algorithm is unsuitable.

Filling-in the missing data with a known value like 0 is certainly incorrect, since in general some of the students that didn't attempt an item would have answered it correctly (in fact, hopefully most of them would). There are, however, a number of possible solutions to this. In this paper, we consider two: alternative optimization and estimation of missing data.

Alternate Optimization. An alternative to optimizing the Euclidean distance metric is presented in (Lee & Seung 1999) where the objective function is

$$F = \sum_{i=1}^{m} \sum_{u=1}^{n} [M_{iu} \log(WH_{iu}) - WH_{iu}]$$

This objective function is derived by interpreting the NMF process as a method for constructing a probabilistic model where value $M_{i,j}$ is generated by adding Poisson noise to $WH_{i,j}$ and then finding the maximal likelihood W and H for generating the known M.

The benefit of using this objective function is that its corresponding update rules

$$W_{i,a} \leftarrow W_{i,a} \sum_u \frac{M_{i,u}}{WH_{i,u}} H_{a,u}$$

$$H_{a,u} \leftarrow H_{a,u} \sum_i \frac{M_{i,u}}{WH_{i,u}} W_{i,a}$$

no longer contain matrix multiplication operations on M. To avoid degenerate conditions, the additional constraint that each column of W is normalized after each update round is imposed.

In the summations, simply skipping those entries that are undefined because of missing values in M is equivalent to assigning 0 to the missing entries. An option that works better is to normalize each summation by the number of missing entries in that summation (thus, multiply by 1.5 if only 2 of 3 values were present). This is equivalent to a pre-processing step that constructs two versions of M: one where missing values are initialized to the average value for the row (a student is predicted to do as well on missing items as they did on average for existing items) and one similarly for columns (a student is predicted to do as well on a missing item as the average of those classmates that attempted the item.) The first version is used for the first update rule (for W) and the second version is used for the second update rule (for H).

Estimate Missing Data. Given this view of the solution, a natural next step is to predict the missing values in preprocessing M before running an out-of-the-box NMF algorithm. Knowing the domain from which the dataset comes can be a great help in overcoming the problem of missing data. Within educational statistics one of the most powerful statistical frameworks is Item Response Theory (IRT) (Baker 2001), the framework behind many computer-adaptive tests such as the new GRE (Wim J. Van der Linden 2000). Although IRT requires larger student populations than are generally found in a single course, some of the concepts from IRT are useful for developing a simple model for modeling question difficulties and estimating missing score data. IRT parameterizes a cumulative distribution of the probability a student with a given level of ability will get a given item correct. Usually these "characteristic curves" are represented with a two-parameter logistic function

$$P(\theta) = 1/(1 + e^{\alpha(\theta - \beta)})$$

where β governs the difficulty of the question and α governs the discrimination of the question. Here discrimination is related to how likely it is that a student with ability less than β can answer correctly and how likely a student with ability greater than β may answer incorrect.

Even without a sufficient population to estimate α and β directly in this model, the concepts of item difficulty and discrimination as a parameterized model are useful ones. A simple technique for estimating α and β is to sort the student responses to the given item j by the student's course grade (as assigned by the instructor) and find the split point β that maximizes the number of students with wrong answers below the split plus the number of students with right answers above the split. We let α denote the fraction of students that β predicts correctly — formally:

$$\alpha = (\sum_{i:S_i<\beta} \delta(M_{i,j}) + \sum_{i:S_i \geq \beta} \delta(M_{i,j} - 1))/n$$

In this view, α is the empirical odds that a student with $S_i < \beta$ will get the item correct or one with $S_i \geq \beta$ will get it incorrect.

It is a simple matter then to calculate these empirical α and β values for each item based on the students that attempted that item. For every other student, we can provide a binary estimate of their score on the item by comparing their "ability" (course score) S_i with the difficulty of the question[1]. If $S_i \geq \beta$ then we fill in $M_{i,j} = 1$ with probability α and $M_{i,j} = 0$ with probability $1 - \alpha$. For scores less than β, we do the reverse.

This is an extremely basic parameterization, but one that takes into account a simplified model of student assessment. When we evaluated with leave-one-out cross-validation on three gradebook datasets, this method provides an accuracy of 63% to 69% when predicting individual responses, depending on the gradebook. Other simple techniques found in the general literature perform worse by as much as a factor of 2, as seen in Table 1.

NMF With Known Values

A simple method to extend the NMF algorithm is to query the instructor for some portion of the question relevance matrix, H. While identifying the entire matrix is time consuming and tedious, providing a fraction of the entries of the relevance matrix can be done quickly. Some intial "seeding" of the relevance matrix with known values can also save post-processing time by eliminating the need to later label the resulting topics with meaningful names.

In our experiments, we label this extension NMF+K along with the percentage of relevance matrix entries provided by the instructor.

Sigmoidal Factorization

The implicit model given by NMF is that student ability affects their score outcomes linearly. A more common model would be a sigmoid, as is commonly used in item-response theory. Applying techniques from (Gordon 2002) allows us to assume a model $M = f(WH)$ where M is the score matrix, W and H are the student ability and question relevance matrices, respectively, and f applies the sigmoid operation to each component of its input matrix and outputs the

[1]The majority of items in our datasets come from multiple choice questions and thus have no partial credit. For datasets with a large population of non-binary items, this technique may not be suitable.

(same-dimensionality) result matrix. This may match more accurately the mental model governing answering items correctly. This algorithm is also solved using a stochastic gradient descent. We present results for this algorithm abbreviated as "GGLLM."

Principal and Independent Component Analysis

In our results section, we also compare to Principal Component Analysis (PCA) and Independent Component Analysis (ICA) (Hyvärinen 1999). Both factor the matrix M into two components, W and H, just as NMF. However, neither require the components to be non-negative. PCA finds the factorization that minimized the squared error in reconstruction. This is also the factorization that renders each factor statistically uncorrelated. ICA finds the factorization that renders each factor statistically independent.

Common-Factor Analysis

In psychometrics,[2] the textbook method for performing topic filtering/clustering is "factor analysis" (Spearman 1904). Originally developed in 1904 by Charles Spearman to support his theories on intelligence, factor analysis combines a number of techniques based on singular-value decomposition of the correlation matrix of the values in question. Unlike the collaborative filtering algorithms discussed previously, factor analysis does not yield a matrix of student ability in each "topic" (without performing a separate factor analysis on the correlation of student scores.) Instead, it produces the matrix of questions' factors without producing the corresponding students' factors matrix.

During its century-long evolution, factor analysis has acquired a large number of standard options and related algorithms. So, we chose to use an off-the-shelf implementation(SAS 2005).

One minor complexity in using factor analysis on educational data is that the correct method of calculating the correlation matrix would be to drop any row of M (student) that has any missing values. Depending on how many instruments are given out during a course offering, this could easily leave no data with which to calculate correlation. The accepted method for dealing with this is to leave out any rows that have missing data in the two columns whose correlation is being measured at any given point. So long as the missing values are randomly distributed (i.e., whether a score is missing is not related to the ability of the student on that question) this will not bias the correlation matrix significantly, but the possibility should be noted.

Probabilistic Relational Models

NMF and CFA look for linear decompositions of the score matrix. We have also used Bayesian networks to learn nonlinear probablistic decompositions. We hypothesize a hidden discrete variable for each question (q_i for question i) and for each student (s_j for student j). The former represents the type of the question and the later the type of the student. These two types of hidden attributes need not align or carry the same symantic meaning as each other.

Our probabilistic model now has three distributions: $p(q_i)$ (the prior probability of the question type), $p(s_j)$ (the prior probability of the student type), and $p(m_{ij} \mid q_i, s_j)$ (the probability that a student of type s_j will receive a score of m_{ij} on a question of type q_i). Together this forms a Bayesian network with one node (variable) for each student, question, and score. The student and question nodes have no parents. Each score node has one question and one student as parents. This forms a probabilistic relational model (PRM): a Bayesian network built from a template (or schema) of shared probability distributions. (Poole 1993; Ngo & Haddawy 1996; Koller & Pfeffer 1998; Friedman *et al.* 1999)

Our data is partially observed: we know the value of the m_{ij} variables but do not observe the q_i or s_j variables. We employ expectation-maximization (EM) (Dempster, Laird, & Rubin 1977) to learn the three distributions above. At each iteration of the EM algorithm, we must calculate the marginal distributions over the variables, given a particular model. Exact marginal computations in this network are intractable. Instead, we use the standard approximate inference method of loopy belief propagation (the application of the polytree inference algorithm of (Pearl 1988) to graphs with loops). Although such a method is approximate and stochastic in its results, our tests with these datasets have shown that the results are consistent and provide stable convergent results.

Once learned, the model can be used to calculate the posterior distributions over the question types given the scores. This is another marginal computation and again we use loopy belief propagation. We assign each question to the type (or topic) with the highest a posterori probability.

4. Preliminary Results

To ensure that our method for filling missing data performed at least as well as any simpler concept we've found in the literature, we tested against the obvious solutions: missing item scores are assumed to be wrong (0), assumed to be right (1), or assumed to be correct with probability equal to the average of the non-missing scores for that student. In each case, our prediction is better, which is unsurprising considering the added expressiveness of the model. The numbers presented in Table 1 are calculated by performing leave-one-out cross validation on the known data in the dataset.

There are two prime methods for evaluating the ability of these algorithms to extract valid topic information. First, we can compare the topic groupings (as determined by the largest-absolute-weight factor for each item) algorithmically generated with groupings created by human instructors given the question text. Secondly, when a full question relevance matrix is provided we can directly compare with that. In this case, we can evaluate the element-wise Euclidean distance between the provided matrix and the generated one.

For topic-grouping evaluation, we had 3 datasets readily available to us. To determine the topics that should be discovered, we asked three instructors for each dataset to make groups of the items by topic. Just as with our algorithms,

[2]The branch of psychology and education concerning test validity and educational statistics.

Method	Avg. Squared Error
Diff+Disc	30.9%
Wrong	63.2%
Right	32.1%
Student Avg	41.6%

Table 1: Comparison of data-filling methods. "Diff+Disc" is our method based on IRT. "Wrong" corresponds to predicting a student will score 0 on every missing value. "Right" corresponds to predicting a student will score the maximum on every missing value. "Student Avg" corresponds to using a student's average score (taken across present values) to fill in missing values for that student.

not all questions needed to be grouped, and some questions could be in multiple groups. Our only request was that they be grouped into topical clusters.

The first was from a recent offering of our undergraduate Operating Systems course (CS153). This dataset has 34 questions on 4 instruments, and 64 students. 28 of the questions are binary. For evaluating the $NMF + K$ extension, 22% of the possible relevance entries were provided by the instructor.

The second dataset comes from an upper-division Algorithms course (CS141). This dataset has 2 instruments, 15 questions, 19 students, and a full relevance matrix provided by the instructor. None of the questions are binary.

The third and final dataset we evaluated comes from an introduction to computing course (CS008). This dataset has 80 binary questions and 269 students. No relevance information was provided by the instructor.

To determine how many factors to extract from our CS153, we applied both the Kaiser criterion (Kaiser 1960) and Cattell's scree test (Cattell 1966), techniques from psychometrics developed to help determine the number of factors to look for. The Kaiser criterion keeps a number of factors equal to the number of eigenvalues of the correlation matrix which are greater than 1. In our case this indicated dozens of factors, which would make the results extremely difficult to evaluate, and was questionable since few of the eigenvalues were significantly above 1. Instead we utilized Cattell's scree test. This test looks for an "elbow" in the graph of sorted eigenvalues of the correlation matrix vs. their rank. It indicated that 5 would be a good number of factors.

For CS141, the relevance matrix was provided for us, dictating the number of factors ahead of time. For CS008, we chose 7 factors based on the number of non-singleton groups identified by our volunteer examiners.

Each algorithm gives a weight relating the strength of the connection between each question and each topic (e.g., the values of the question-factor matrix H for NMF and the probability that the posterior distribution of the type of the question given the data for PRM approach). Each algorithm was run to find factors and output the relevance of each item to those factors. Each factor's four highest-weight items constituted its *selective group*. But, from each selective group, we omitted every item whose weight was less than 33% of the maximum output — that threshold was chosen because empirically it removed very few items but prevented any item from belonging to more than one group. Additionally we identify the *full group* for each factor as the set of items that have more relevance for that factor than for any other factor. Thus, every item is placed into exactly one full group.

Our results are presented in Tables 2-4. We looked at four statistics for quantifying how well these algorithms discovered factors corresponding to the human notion of topic:

- *Selective Precision* - For the selective groups described above, how many of the intra-group pairings can be found in the groups identified by a human? For example, if q2 and q7 are placed in the same selective group, are q2 and q7 found in the same group in any of the human-created groupings? If an algorithm returns the trivial "no groups" result, precision will be 100%.
- *Selective Recall* - For each intra-group pairing in the human-generated groups, can that pairing be found in the selective groups identified by the algorithm? This is in some sense the inverse of the above. If an algorithm returns the trivial "everything in one group" result, recall will be 100%.
- *Full Precision* - Precision as described above, but on the full group.
- *Full Recall* - The recall of the full group.

Selective groups vs. full groups can be viewed as two points along a continuum. In addition to providing these two extremes, Figure - show the entire precision-recall curves for NMF, SVD, CFA, and PRM on all three datasets. We only present results on these four for clarity, and selected these because they are the most diverse of the set of evaluated algorithms.

Since not all selective groups have four members and since full groups have varying sizes, the number of possible pairs that match varies from grouping to grouping. Thus, rather than the raw number of matches, the more meaningful percentage agreement is provided. Since the NMF, GGLLM, and PRM algorithms are stochastic, we average these figures over ten runs and provide standard deviations.

As a baseline comparison, we also show results for the average of evaluating ten randomly generated groupings. All statistics are presented in Tables 2-4. Note that since the CS008 dataset comes from a single exam, there are no missing entries, and thus no need for data completion.

Our second method of comparing the effectiveness of the various algorithms is to examine how well they can reproduce a relevance matrix from the data. For our CS141 dataset, we were kindly provided a full relevance matrix, encoding at a more precise level the instructor's domain knowledge in terms of identifying similar questions. Figure shows the decreasing error for the NMF+K algorithm as the supplied percentage of the relevance matrix increases. Table 5 shows average reconstruction error of the provided relevance matrix for various algorithms. Again, for stochastic algorithms results are averaged over ten runs and presented with one standard deviation.

Algorithm	Data Completion	Selective Precision %	Selective Recall %	Full Precision %	Full Recall %
NMF	N	29 ± 6	5 ± 1	29 ± 2	25 ± 5
NMF	Y	32 ± 9	6 ± 1	29 ± 2	27 ± 4
NMF+K(22%)	N	29 ± 8	5 ± 1	30 ± 2	28 ± 6
NMF+K(22%)	Y	35 ± 6	6 ± 1	30 ± 3	29 ± 5
FastICA	Y	29.6	5.3	29.1	22.0
SVD	Y	46.7	11.0	29.7	24.4
GGLLM	Y	30 ± 7	5 ± 1	27 ± 4	23 ± 5
PRM	N	23 ± 5	3.6 ± 0.7	26.8 ± 0.3	26 ± 2
CFA	N	30.0	5.7	27.5	18.7
Rand	-	30 ± 10	4.5 ± 2	30 ± 3	20 ± 3

Table 2: Comparison of filtering algorithms, CS153 Dataset

Algorithm	Data Completion	Selective Precision %	Selective Recall %	Full Precision %	Full Recall %
NMF	N	50 ± 10	17 ± 5	36 ± 5	27 ± 9
NMF	Y	46 ± 7	13 ± 1	36 ± 3	27 ± 4
NMF+K(33%)	Y	40 ± 10	16 ± 5	37 ± 5	32 ± 20
NMF+K(66%)	Y	44 ± 9	15 ± 3	41 ± 6	30 ± 8
NMF+K(100%)	Y	100	28	57	59
FastICA	Y	38	20	40	22
SVD	Y	48	22	46	33
GGLLM	Y	34 ± 9	12 ± 4	40 ± 5	40 ± 10
PRM	N	25 ± 0	6.5 ± 0	44.2 ± 0	41.3 ± 0
CFA	N	42	22	32	15
Rand	-	40 ± 10	10 ± 4	37 ± 6	24 ± 6

Table 3: Comparison of filtering algorithms, CS141 dataset

Algorithm	Selective Precision	Selective Recall %	Full Precision %	Full Recall %
NMF	27 ± 5	1.2 ± 0.5	35 ± 2	27 ± 4
FastICA	28.6	0.8	32.8	25.2
SVD	38	2	35	30
GGLLM	30 ± 20	0.3 ± 0.6	28 ± 1	50 ± 10
PRM	50 ± 4	1.9 ± 0.5	28.6 ± 0.9	27 ± 3
CFA	29.7	1.4	25.4	12.2
Rand	26 ± 7	1.3 ± 0.7	28 ± 2	14 ± 1

Table 4: Comparison of filtering algorithms, CS008 Dataset

Algorithm	Data Completion	Percent Error
NMF	N	0.15 ± 0.02
NMF+K(33%)	N	0.12 ± 0.02
NMF+K(66%)	N	0.09 ± 0.01
NMF	Y	0.13 ± 0.02
NMF+K(33%)	Y	0.14 ± 0.02
NMF+K(66%)	Y	0.12 ± 0.02
SVD	Y	0.17
ICA	Y	0.13
CFA	N	0.17

Table 5: Average relevance matrix reconstruction error

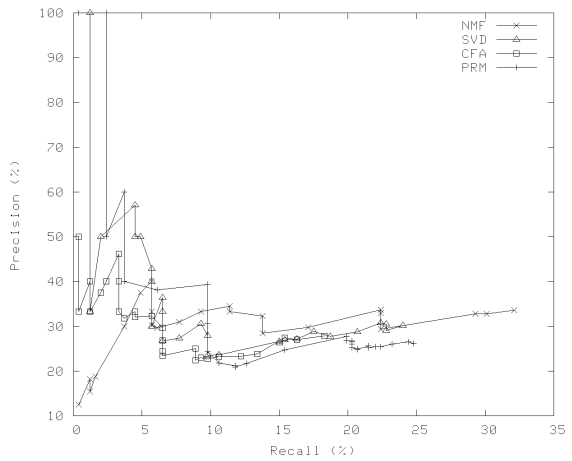

Figure 1: Precision-Recall of Selected Algorithms, CS153 Dataset

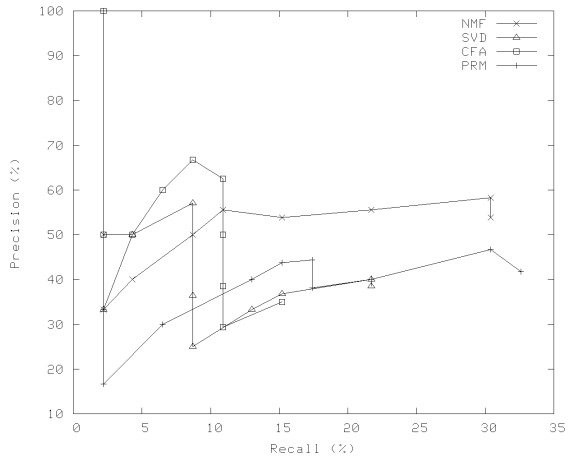

Figure 2: Precision-Recall of Selected Algorithms, CS141 Dataset

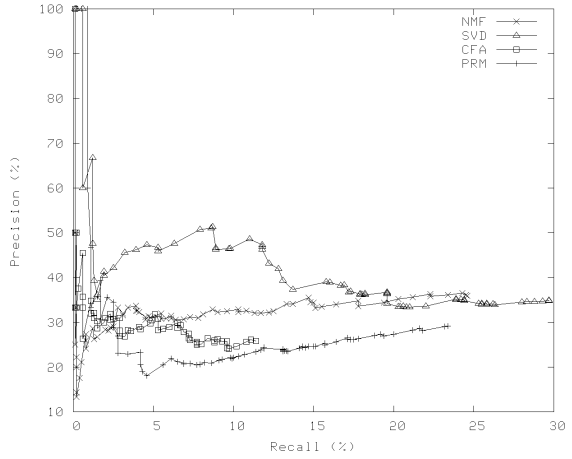

Figure 3: Precision-Recall of Selected Algorithms, CS008 Dataset

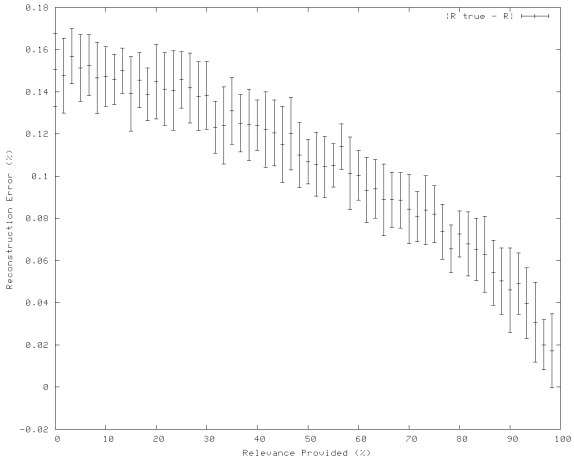

Figure 4: NMF+K Error vs. Percent Relevance Provided

Discussion

On these datasets, using the topic-grouping evaluation, only SVD consistently outperforms a random grouping. The various implementations of NMF perform better on the non-binary CS141 dataset, but there is insufficient evidence to draw strong conclusions from this.

One algorithm whose performance needs to be specifically examined is that of CFA. While CFA is the textbook method for performing precisely this task, its performance is mixed at best. For both selective precision and selective recall, CFA generally ranks above average. On the opposite end of the precision-recall curve, CFA performs worse than random on every dataset presented.

It is also notable that the data-estimation technique provides improvement in the precision when evaluated under topic-grouping, but the opposite appears to be true when directly evaluating the reconstruction of the relevance matrix. This suggests that other data mining forays into education should be mindful of educational models of learning and cognition when building specialized models for a given problem.

When evaluating the algorithms under the reconstruction test SVD, previously our hands-down winner, performs as bad as CFA: to a statistically significant level, these algorithms perform worse numerically. The reason for the disconnect between performance on these two tests is as-yet unknown. Given that we currently only have one full relevance matrix, it is too early to generalize these results. Certainly the numeric performance of NMF here merits continued investigation with this family of algorithms, unlike ICA and CFA which we feel can be safely ignored in future evaluation.

5. Conclusions

The practice of data mining on item-level data is confounded by a number of factors. Most notable amongst them is the level of randomness in seemingly stable scores. A concept that is well-known in educational statistics is *test reliability*,

the level to which re-administering the test would result in the same scores.[3] In our experience, instruments given at the University level are often thrown together haphazardly, but scores are treated as if they were precise values rather than a single high-uncertainty measurement. Thus, the level of "noise" in the average gradebook dataset is at times staggering. Techniques that do not take into account the underlying mental models are unlikely to succeed at extracting any information from this data.

However, there are data-analysis techniques from the fields of educational statistics, data mining, and psychometrics that can wring useful information from these datasets of per-item scores. At this point, the level of precision of the results still remains too low to build usable tools using this technique, but the results are promising. A number of improvements have been proposed, but the question becomes, "What is the minimum amount of information necessary to extract topic clusters from item-level scores?" We feel that the results presented here demonstrate feasibility, but educational aids based on this technology are not yet forthcoming.

6. Future Work

Significant work remains in this area. Our goal is to be able to correctly extract topic information with as little input from the instructor as possible. If item-level scores and course grades are insufficient to get accurate results, as our investigations to date have shown, we plan to investigate how much prior knowledge of question and topic information is necessary to get accurate clusters.

Alternatively, investigation into whether the statistical models that are being generated are in agreement with psychological models of cognition and learning could resolve the discrepancy: it may simply be that the topic clusters that are being provided by our volunteers do not match (from a constructivist perspective) the models assembled by students. This seems likely, as the various algorithms for extracting latent variables seem to have only weak correlation with instructor-identified notion of topic.

Acknowledgments

We would like to thank Dr. Mike Erlinger for taking the time to cluster our operating-systems questions and Dr. Neal Young for the full relevance matrix.

References

Baker, F. B. 2001. *Fundamentals of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation.

Cattell, R. B. 1966. The scree test for the number of factors. *Multivariate Behavior Research* 1:245–276.

Dempster, A.; Laird, N.; and Rubin, D. 1977. Max likelihood from incomplete data via EM. *Journal of the Royal Statistical Society B* 39(1):1–38.

Friedman, N.; Getoor, L.; Koller, D.; and Pfeffer, A. 1999. Learning probabilistic relational models. In *IJCAI: International Joint Conferences on Artificial Intelligence*, 1300–1309.

Goldberg, D.; Nichols, D.; Oki, B.; and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35(12):61–70.

Gordon, G. 2002. Generalized^2 linear^2 models. In *Advances in Neural Information Processing Systems*.

Hyvärinen, A. 1999. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks* 10(3):626–634.

Kaiser, H. F. 1960. The application of electronic computers to factor analysis. *Educational and Psychological Measurement* 20:141–151.

Koller, D., and Pfeffer, A. 1998. Probabilistic frame-based systems. In *Proceedings of the American Association for Artifical Intelligence*, 580–587.

Lee, D. D., and Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788–791.

Lee, D. D., and Seung, H. S. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, 556–562.

Ngo, L., and Haddawy, P. 1996. Answering queries from context-senstivie probaiblistic knowledge bases. *Theoretical Computer Science* 171:147–177.

Pearl, J. 1988. *Probaiblistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Poole, D. 1993. Probabilistic Horn abduction and Bayesian networks. *Artifical Intelligence* 64:81–129.

SAS. 2005. Sas 9. http://www.sas.com/software/sas9/.

Spearman, C. 1904. General intelligence, objectively determined and measured. *American Journal of Psychology* 15:201–293.

Wim J. Van der Linden, C. A. G., ed. 2000. *Computerized Adaptive Testing: Theory and Practice*. Kluwer Academic Publishers.

[3] Assuming of course that nobody learned anything from the first test.