Masaryk University
Faculty of Informatics

Techniques for measuring similarity of educational items

Bachelor's Thesis

Dominik Gmiterko

Brno, Spring 2018

This is where a copy of the official signed thesis assignment and a copy of the Statement of an Author is located in the printed version of the document.

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Dominik Gmiterko

Advisor: Radek Pelánek

Acknowledgements

These are the acknowledgements for my thesis, which can span multiple paragraphs.

Abstract

This is the abstract of my thesis, which can span multiple paragraphs.

Keywords

similarity, metrics, programming, keyword2, …

Contents

Introduction

Tutoring systems are computer-based systems designed to introduce users into various domains. They usually hold a large number of items which enables them to provide a personalized experience. To maintain this large pool of items efficiently we need to be able to decide which items are useful and which are not.

Besides Introduction and Conclusion chapters, this thesis is structured into three additional chapters. First chapter talks in general about the problem of measuring similarity of educational items, advantages of using similarity. This chapter also describes previous work and different proposed metrics for computing similarity. Later we explain different types of data available in tutoring systems. The second chapter then advances level deeper and describes observed problems specific to data we are using. The last chapter gives an overview of many experiments that were concluded and summarizes results.

1 Similarity

In this chapter, we talk in general about items (questions) in tutoring systems and computing their similarity. In the beginning we of the chapter we focus on better explaining basic terms like tutoring system, item, and similarity which will be used all throught the work. We also describe different kinds of data available about items and possible metrics for calculating similarity. The last section of this chapter describes data used in our experiments. In particular, we describe used dataset and outline ways of simulating artificial data.

1.1 Tutoring systems

Tutoring systems are computer systems which purpose is to teach its users (students) some knowledge or skill. These systems do this autonomously to some degree. The degree of automation may vary for each tutoring system.

Many different aspects of learning may be implemented into the system or they may also be left out. The most common task which tutoring systems solve is choosing a most beneficial item from the pool of items available for users to solve. This can be done in some simple way or system may also adapt to each of the users choosing more challenging and interesting items to users who are more skilled in a specific area. While presenting users who struggle with more basic items. This is beneficial for maximizing learning aspect of the system. [CITE?] Another aspect which can tutoring systems solve is providing hints and feedback to students. When there are teachers interacting with the system we can also provide feedback about their students progress and so on.

One closely related area are recommender systems which differ from tutoring systems only slightly. Both areas are observing users and items. The main difference is that our main source of information is a performance of users while solving specific item and recommender systems mostly use rating of the items. Which in fact are still only numbers representing slightly different things so we can still share some of the techniques. Although different goals bring different problems specific to each area.

One difference is that we can use some additional data about items. There are item statements and solutions of students which can in some contexts give us really useful information about items.

1.2 Items

In this work, we use the term "items" which may refer to problems, questions, assignments in different systems. Item is a single entry in an educational system which users can answer to. Since many aspects of this work are generally applicable we decided to use this broad term. Complexity of single item in tutoring system can differ greatly. In some tutoring systems, this may refer to simple choice from two options while in other one item is complex tasks which user solves in a matter of minutes.

To further specify the context of our research, we will describe characteristics of items. Computing similarity may be performed by different metrics, but they all have to use data which are available for each item. Therefore we first describe sources of data that can be utilized for measuring the similarity of items.

- **Item statement** is some specification of the item for a learner to solve, e.g., a natural language description of the task. Another commonly used format is a grid. Many systems focusing on with logic puzzles and programming problems use it.

- **Item solutions** may provide us with additional information about an item. There will usually be some sample solution provided by the author and we can also utilize all the solutions from users.

- **User's performance** consists of information provided indirectly by users. Performance of items may represent user solving times, correctness of the answers, number of attempts required.

Structure of both item statement and solutions differ greatly based context of tutoring system. However, in general, it is still possible to convert data into some standard form and use few standard metrics despite original representation.

This description of an item is broad enough to cover most of the tutoring systems. End of this chapter discusses more closely tutoring system used for our results.

The most important property of tutoring systems is that they contain educational items which are solved by users. However, amount of items in systems may vary. Some tutoring systems require a large number of items. This is especially true when they practice simple facts instead of skills. In personalized tutoring systems, the need for a large pool of items is even higher. When providing personalized questions we need a wider set of items to choose from.

Dealing with a large pool of items may not be easy. To maintain a pool of items efficiently we need to be able to easily decide which items are useful and which are not. One possible tool which can help us here is similarity of items.

1.2.1 Measuring similarity

Following section is explaining general approach to measuring and using similarity of educational items.

In general, we can compute similarity of items in many ways. We define metrics for computing similarity for two items. Then when we want to use similarity for some specific use-case we compute pairwise similarities for all pair of items.

Items in our context are commonly represented as vectors of numeric values. For performance data, this is vector of correctness or time from all users to given item. Other properties of items like question statement may also be represented as a vector, e.g., by using bag-of-words.

When we have vectors for each item we can compare them pairwise to get similarity using some standard similarity measures like Pearson correlation coefficient, cosine similarity, Sokal measure or Euclidean distance.

Another possible way to measuring similarity is counting edits which would convert one item to another. This is called edit distance and there are standard ways of computing it for both strings and trees. So this covers another common group of information we can encounter.

1.2.2 Elements of standard pipeline

So we can wrap it up. There are a few data structures and few calculations involved in the standard pipeline for computing similarity of educational items.

1. **Feature matrix** is matrix (items $\times$ features) containing source data. As we said previously this can represent any property of the items. This matrix consists of one vector for each item and it may represent item statement, solution or solutions of the item and naturally even users performance.

2. **Measuring similarity** may involve some similarity measure or edit distance. This step is used to compute the similarity between all pairs of items. In other words, we transform feature matrix into similarity matrix.

3. **Similarity matrix** is matrix (item $\times$ item) where each value represents similarity of pair of items.

4. **Dimensionality reduction** is used to transform similarity matrix into a projection. Techniques like PCA or t-SNE may be used for this.

5. **Projection** is more compact representation (item $\times$ 2) of similarity matrix used for visualizations for end users.

1.2.3 Why is similarity of items useful

As we mentioned previously key part of learning is solving educational items. Defining some metrics for measuring the similarity of items can then be used for different purposes.

First, most direct, usage is a recommendation of items for a student to solve. We do not want to recommend very similar items to those that were solved without any problems. However when user struggled system should consider recommending more of the similar problems to strengthen users knowledge.

Another possible usage is generating hints by selecting examples which are similar to the item which is currently solved. Examples are selected from a database of examples. This usage of similarity was used by [CITE Hosseini; Brusilovsky].

Two previous use cases were using problem similarity to automatically make some choices inside tutoring system. Another approach is to bring humans into the decision-making loop [CITE]. This approach provides authors of tutoring system with visualizations which should inform them what changes may be useful. E.g. detecting possibly redundant problems or pointing out where there are not enough similar problems to choose from user struggle.

One possible way of achieving this is plotting problems to plane and displaying it to the author. This still can not be used for a very large amount of items. But we choose this approach for our specific data as they contain a large number of items but can be divided into item-sets which are solved independently in the system.

One of the use cases of visualizations is a detection of outsiders - items which behave differently than others. This behavior is directly shown in projection and such items lie far away from others.

There is one more usage of similarity which we won't be discussing further. The similarity of items can be utilized for automatic construction of clustering and hierarchical categorization. Even when author already has items categorized he can compare it to computed categorization to verify that groups are formed correctly and refine them if needed.

2 Used data and techniques

In our analysis, we use both real data from the educational system and simulated data. There is a reason why use both as only real-world data are useful for concluding any practical results. However, evaluation of this data is often complicated as we do not know the truth about many of their aspects. That is why we use simulated data for validating some of our conclusions.

2.1 Umíme česky

Umíme česky is a system for practice of Czech grammar. The system contains multiple exercise types, but in our analysis, we use only one exercise type - simple "fill-in-the-blank" with two possible answers. A user is asked to choose one of them.

We focused only on "fill-in-the-blank" exercises although they can still be used to train various concepts of Czech grammar. The exercise contains many questions but they are divided into item-sets. Each item-set consists of items practicing a different aspect of language, e.g., "Vyjmenovaná slova po B" or "Velká písmena: státy, oblasti". This item-sets are arranged into hierarchical categorization as seen in the user interface.

Items in item-sets are then divided into levels. Where higher levels are intended for solving by more experienced users as they contain more difficult questions. In general, there are three difficulty levels but not all item-sets have all three of them. Some easy item-sets include only first level. On another hand, there are item-sets which have all three levels or only higher levels.

The used dataset contains multiple sources of information about items. In "item statement" group it is statement of the question with one missing spot and two possible answers to fill in there. We also know answers from all the users. Then in performance group of information, it is correctness of user answers and response time.

It is not possible to use response time directly. We need to normalize it in some clever way as raw response time is greatly affected by both lengths of questions and users reading speed. Also, it is a good idea to use logarithm of time instead of time itself. This is shown by [CIET].

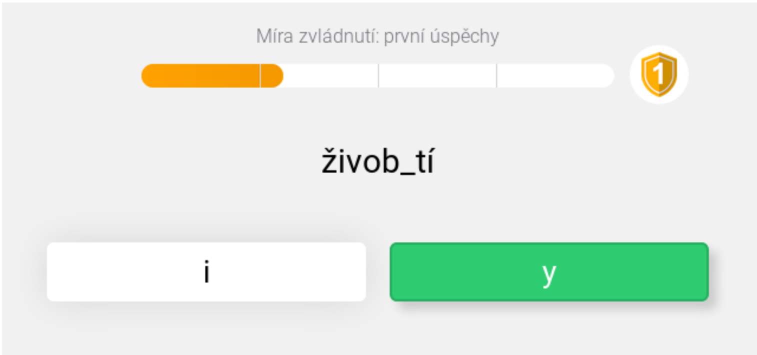

Figure 2.1: "fill-in-the-blank" example question

Only then it would be useful to use data about response times. This is a reason why we are using mostly correctness of user answers.

We choose this tutoring system because, as we explained before, we are solving problems which are most important for systems with a large number of items. Also provided dataset has a large number of users and answers which is great for the stability of results.

2.1.1 Basic statistics

	Items	Users	Answers	Item-user answers
"Fill-in-blank"	6 037	46 128	10 421 521	7 264 763
One item-set	273	14 207	1 216 403	888 748

The last column contains unique item-user pair answers. This number differs from all logged answers as some users may have answered some item multiple times. However, we use only one of the answers.

One selected item-set is the one most commonly used for analysis. This item-set has most answers from users, therefore, it is ideal for analysis as results are more stable. We used this item-set by default but we also confirmed observed behavior on other item-sets.

The primary group of users using Umíme česky are children aged ?? to ??. The system also supports teachers and usage in school. Each week about 200 classes visit the system.

2.2 Simulated data

In addition to the real dataset, we also used simulated data. This section will describe how we simulated answers to items.

The result of the simulation is performance matrix with n items as columns and u users as its rows. First, we have to create items. For each item we choose its difficulty, skill required to solve this item. The difficulty is value drawn from normal distribution $\mathcal{N}(0, 1)$.

After that, we continued with the construction of users. We generate skills for all users - a matrix of u users and all used skills. This matrix also contains random values from normal distribution $\mathcal{N}(0, 1)$.

Next, we simulate each user answering to each item. User answers correctly whenever is a random value higher than a logistic function of the difference of item difficulty and user skill. When it is not given answer is incorrect. User skill is one of the skills which corresponds to the skill required for given item.

We will talk more in depth about how we generated simulated data in next chapter when describing how we used them specifically. In all experiments using simulated data, we altered this basic simulation in some way to achieve results corresponding to some attribute of real data.

2.3 Previous work

Dataset from Umíme česky was used previously in few articles and thesis.

- -

It was determined which similarity measures it is best to use for stability of results compare similarity measures to each other some metrics correlate more that other using Pearson is good default choice

2.4 Projection

Projections come in handy when it is hard to understand data directly because there is too much of them. This is especially true for tutoring systems which consist of thousands of solvable items and even more users in cases like this it is not possible to look at data about each item

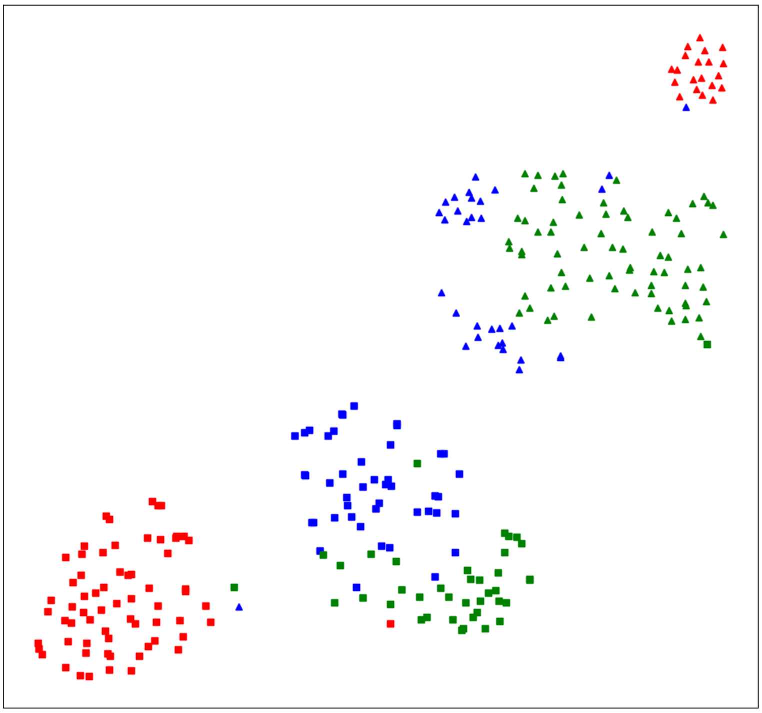

Figure 2.2: Basic projection of one knowledge component in Umíme česky

individually possible with projection of many-dimensional data into 2 dimensions.

In general we want our projection to put similar items together. This can be achieved in many different ways. It is important to choose correct source of data and method of processing them prior to applying dimensionality reduction.

Figure 2.2 shows how common projection looks like. This particular projection shows 273 items of single concept from system. Each item is represented as one dot in the image and its proximity to others represents how similar they are.

2.5 Visible Properties of projections

we want to use results of calculated similarity and projections for managing item pool. That puts some constrains on how ideal results look like.

We want similar items to end as close to each other as possible. When we use performance as data source this means items which require same skills are projected near each other. For example when we look at our item-set we can see that all words words starting with "bio" are close to each other. This group consists of words witch are indeed similar because they are based on single word with added suffix. However there is no information about item statement presented to similarity metric.

After looking at image you can notice some regularities.

2.5.1 Computation of item similarity

To better understand how was this image created we have to understand how it is computed. We will describe it following section. This is default work-flow we used in most cases. Whenever we don't specify otherwise all projections were produced using this work-flow.

First step we have to do is converting raw logged information about user answers to **performance matrix**. This matrix consist of entries for each user-item pair. Columns of the matrix are items in educational system and each row of the matrix contains data about single user's performance. In most cases we used correctness of first users answer to specific item as his performance. This means value 1.0 in case of correct answer and 0.0 for incorrect. Another possible choice is to incorporate user solving time into this value [CITE]. Performance matrix is relatively sparse as it is not common for users to solve all the items in the system.

We tried using both first users answer and last in performance matrix. However there is no visible difference in resulting projection (same clusters are formed). This is clear when we compare first and last answers of users. There is only 5% difference between them. So at least with large number of items in system it does not really matter which one is chosen.

Next step is computing **similarity matrix**. It is filled with pairwise correlation of all items from performance. We choose Pearson correlation coefficient for this and it was used for several reasons. Previous work showed that it has good enough results in producing distinct clusters on performance data [CITE]. Same article also showed that using second level improves this property even further. However, we choose not to use it to simplify explaining our experiments as that is important factor to our work. Another reason is that Pearson correlation is easy to use as many computational environments already contain fast implementation.

The last step is producing 2D **projection**. Similarity matrix may be used by computed directly for decision making. However this matrix still contains way too many values for human to interpret. As some of our goals are explaining data to users. We continue with dimensional reduction to produce more compact representation of data. This can be achieved by using any dimensional reduction technique.

We choose to use Principal component analysis (PCA) and there is several reasons for doing so. Result of PCA is deterministic. It produces same result for same input each time it is run. This is not true for TSNA which is technique using machine learning and gradient descent for finding some local extreme. Stable results are more suitable for understanding data as there is much less variation to results caused by algorithm. It is much easier to compare results when altering metrics used for computing item similarities.

First two principal components of PCA are then used for 2D visualizations.

We choose this specific work-flow as we think it is utilizing data about items which hold most information about their similarity. As other possible choices are item statement and solutions provided by students they do not hold as much information. Item statements in our particular exercises consist only of few Czech words. Also student solution is only choice from two provided options. Item statement and solutions can be used more effectively in other contexts like programming, mathematics, physics, or chemistry.

This choice of work-flow is also relatively simple and easy to understand. It consist of few steps which can be studied separately and interchanged.

3 Evaluation

3.1 Level regularity

When you look back at figure 2.2 you can see that there are three colors of items. Most TODOknowledge components in system Umíme česky are spitted into multiple levels of difficulty. For this particular knowledge component there are three difficulty levels. First level is shown with red, second with green and third blue. This shows visible pattern in our data - each color (level) forms a distinct cluster. In following section we will try to explain factors that can affect resulting projections produced from real-world data.

Projected item-set is divided into three levels with 110, 86 and 77 items respectively.

As we mentioned before, only data about user performance (correctness of answers) are used when composing projections. And there is not a direct reason for this clusters of same levels to form as no information about belonging to particular level is presented to the algorithm.

Levels are not solved uniformly. It is not common for user to sole all three levels. Less experienced users tend to solve only first or first two levels. However typically older users solve only higher levels. Main cause for this is that system allows teachers to assign particular level of concepts as homework. Students then usually solve only this single level.

This phenomenon is not suitable for analysis of item similarity as it can cause misleading results. One particular example is when similar word are displayed far away form one another just because they belong to different levels. This is visible for words "bič" and "bičík" in item set "Vyjmenovaná slova po B".

3.1.1 Sparseness of performance matrix

In same way as we explained before in section ??. Only difference is that resulting performance matrix will now contain missing values. We wanted to recreate missing data in similar way that this occurs in real performance matrix. This is achieved by simulated users not solving all items. Each user starts with solving one level and then with some probability continues to another. So most users solve only one level, some users solve 2 levels and only few users solve all 3 levels. Order in which they answer levels is chosen at random as users are not required to continue chronologically. This is also visible in real data - there are users who solve only second or only highest difficulty level.

Results indicate that that structure of data can affect results only in really special cases. The only case where projection is divided into clusters is when there is absolutely no information between question in different levels. But this is not our case in real data as there are users solving multiple levels.

3.2 Users similarity projection

Insead we choose to analyse whether our data contains different groups of users. We are changing how we are looking at data.Up until now we used item-item similarity to calculate projection of items. Now we are going to to be using user-user similarity.

It is worth mentioning we will work with larger matrices as this forced us to do some optimalizations in code of our analysis. For example similariy matrix of items usualy has size between 100×100 and 300×300. As we are looking only on items from single knowledge component. Hovewer user similarity matrix is square matrix with size around 10000×10000. This is also reason why current recommender systems use item similarity instead of user similarity. CITE

First attempt on projecting users can be seen in figure ??.

We say that user solved level when he answered at least 30 questions (which is 1/3 of questions for observed item set). Based on this we can divide users into 8 groups. We added color to each group so we can distinguish them in following plots easily.

Group	Color	Users count	User percent
none	black	4984	35
only 1st level	red	4375	31
only 2nd level	green	1285	9
only 3rd level	blue	1114	8
both 1th and 2nd	yellow	960	7
both 2nd and 3rd	cyan	224	2
both 1nd and 3rd	magenta	240	2
all levels	light gray	1025	7

We can see that all formed clusters of users contains in most cases users from single user group when we divide them by levels they solved. This brought us to more interesting discoveries. In general each level and each item has some mean performance. This is shown in figure ??. Horizontal axis contains items sorted by level and mean performance. Vertical axis shows performance of each item. Given item sets have mean performance 94%, 86%, 71% for each level respectively.

After dividing users into 8 groups plot changes slightly. Some groups like "only 1st level" (its colored red and contains users who solved primarily first level and only few or none items from other levels) has much lower performance on other levels. We can conclude that users who tend to solve mostly first level are not as experienced as other users. On other hand users in group "both 2nd and 3rd" (cyan) are performing better than other users on all three levels.

At this point we have to return to simulation. We want to show that grouos of users solving levelswith different performancecan cause forming of clusters of items from same level.

Answers are simulated pretty mutch same as in previous simulated experiments. We have 300 items divided into 3 levels. There is 3000 simulated users. Most of them solve levels with same skill but some 1/5 of users have smaller chance to solve second and third level. This is visible on item performance polt similar to one from real data.

Only chnged variable was performance of some users for levels and this resulted in visible clusters in projection (figure ??).

From otained information we can conclude that removing subnormal answers of users (few answers to other levels when they solved primarly one level) should remove clusters of levels.

But does it?!?

3.3 Different performance metrics

We applied 4 different metrics for computing similarity matrix to verify that previous results aren't specific to single metric. Especially if it is true that similarity of items differ based on correct answer.

With boolean performance data, we can summarize performance of all users on items i and j just by using four values. Count of users who solved both questions i and j... [TODO cite metrics]

Pearson and yule produce almost identical results - this confirms previous research. This means plots produce same distinct clusters of answers and total similarities. However Jaccard measurement differs in this aspect. Similarity of items is not greater in one group of questions (based on correct answer).

Different methods for computing similairty may measeure diferent aspects of items. This can be seen when using Jaccard an Sokal metrics. SImilarities provided by Jaccard metric differe greatly from other metrics. Resulting projection still shows same clusters of level and items are splited based on correct answer but information is kept in another way than in correlation based metrics like Pearson.

On other hand Sokal is heavily depending on performance of items. Items with higher performance are much more similar than items with lower performance. This is causing packed cluster of first level (easy) and spread out cluster of items from third level (harder).

3.4 Answer regularity

clusters based correct answer (i/y) when there is no information about this in data used for computing similarity

When exploring data, it may be useful to detect outsiders. It may be useful for multiple reasons [CITE]. In our particular case we declare item an outsider when it is not similar to any other items in item set. seems like a logical path to take.

In particular this means item with low sum of similarities to other items may be an outsider. This is where we encountered another regularity in data.

We can quantify whether this is present in all item-sets. We execute this by calculating quality of k-means clustering on items similarity compared to correct clusters. Where correct clusters are formed as combination of level and correct answer. Quality of clustering is evaluated using Rand index. So value 1.0 represents that k-means divides all items into same clusters as correct classification. On other hand lover values represent it is hard to divide items correctly and there are no distinct clusters.

	min	median	max
delka-samohlasek-i	0.272173	0.572914	0.733154
delka-samohlasek-u	0.179007	0.218907	0.278768
koncovky-mi-my-ma	0.173923	0.378383	0.574999
koncovky-ovi-ovy	-0.022328	1	1
koncovky-podstatnych-jmen-muzsky-rod	0.460326	0.526986	0.652672
koncovky-podstatnych-jmen-stredni-rod	0.444935	0.859304	0.880381
koncovky-podstatnych-jmen-zensky-rod	0.358566	0.423671	0.501532
koncovky-pridavnych-jmen	0.431434	0.500458	0.558587
me-mne-samostatne-zajmeno-ja	-0.012182	0.289142	1
predlozky-s-z	0.31752	0.413021	0.455101
psani-be-bje	-0.009319	0.671716	1
psani-me-mne-ve-slove	0.646219	0.733133	0.733133
psani-nn-a-n	0.305013	0.409739	0.462562
psani-s-c-s-z	0.288553	0.353498	0.44766
psani-ve-vje	-0.013296	1	1
shoda-podmetu-s-prisudkem	0.189355	0.245732	0.306245
sklonovani-zajmen-jez-jenz-nimz-ji	-0.01751	0.121347	0.353978
slovesa-podminovaci-zvratna	0.06066	0.10416	0.173317
tvrde-a-mekke-souhlasky	0.135091	0.257167	0.832182
vyjmenovana-slova-po-b	0.385475	0.597867	0.637596
vyjmenovana-slova-po-l	0.575217	0.641585	0.837544
vyjmenovana-slova-po-m	0.478953	0.928013	0.943975
vyjmenovana-slova-po-p	0.641131	0.832391	0.88884
vyjmenovana-slova-po-s	0.373571	0.421781	0.446662
vyjmenovana-slova-po-v	0.483608	0.678045	0.722796
vyjmenovana-slova-po-z	0.528562	0.565708	0.657444

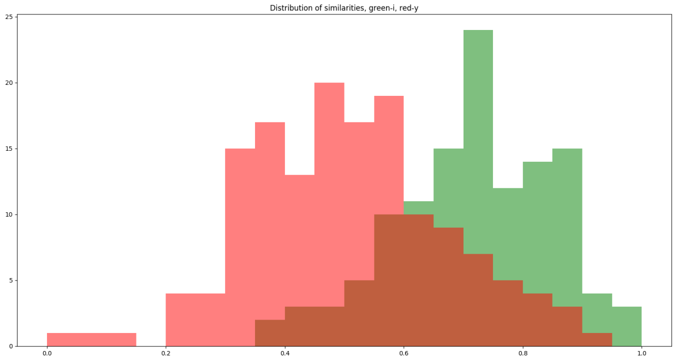

Figure 3.1: Histogram of similarities

This statistic show that many item-sets have distinct clusters of different levels and answers. There is only few sets with obscure structure of answers and this item-sets have large number of possible answers.

Sum of similarities is sum of one column of similarity matrix which we produced in our workflow for computing projection.

This is not specific to only one problem set (which was used in previous images), almost all problem sets display similar pattern. However for some its more distinct than for others. (Sets consisting of items with many possible answers do not behave this way. But that is to be expected.)

Our next experiment was simulation of users preferring one answer in case they do not know what is the correct answer.

Answers are simulated for each user and question. There are no missing values. Answer is correct whenever random value is higher than logistic function of (users skill) - (difficulty of question). Half of questions has one correct answer and other half another answer. We used this to shift chance of answering correctly higher or lower (+0.1 for first answer and -0.1 for second).

Colors represent answer of each item (chance shifted up or down). There are formed visible clusters of same answers in result similar to real-wold data. Using two uncorrelated skills cause clusters in projection. (When using only single skill for all questions results does not change and they are not required for this simulation.)

pridavnych mien je to Y (v pravo)

This is not specific to only one category (which was used in previous images), many more categories display similar pattern. However for some its more distinct than for others. For some distribution of similarities is same for all answers (psani-nn-a-n, vyjmenovana-slova-po-p,..).

We picked few witch have very distinct clusters of answers and looked whether it is answer on left or on right in user interface. Our finding was that not all concept prefer answer which is located

data?

We observed before that whole data set uses both answers the same but for single user this may not be true. There are even users who use only one answer as seen in following performance matrix. There is a line with all questions answered incorrectly for one answer and correctly for another.

We can filter this users out. Next three images show different levels of filtering. First uses no filtering at all (uses all users). Second and third image filter users by difference between performance on each answer. (So if users has same performance on question with both answers this value is 0.0 and when user uses only one of the answers (biggest possible difference) value is 1.0). Second image shows value of 0.3 (performance on one answer is 30third image uses value 6

When we use only half of users which have uniform answers there are no clusters of question with same answers. (Half of users because median is used as filtering value.)

We can look at histogram of difference between answers. For most of the users there is almost no difference but there are users with higher difference in performance between answers.

I look at difference between performance (amount of correctly answered questions) instead of usage of each answer because it is performance that affects calculated similarity of questions. However they should correlate somewhat. Another reason is that it is closer to simulated experiment so we can compare results directly.

When we simulated users we gave them all habit to use one answer more commonly. But not all users have this habit in real-wold data. However there is quite large amount of users who do.

This resolves first problem we encountered but does not explain other regularities in data.

3.4.1 Simulation of default answer

This next simulation is also using same core but is enhanced with further choosing correct answer for each item. This is used to offset logistic function higher or lower to simulate higher chance of succeeding in solving items with correct answer is the users preferred one. We think this is good enough solution to simulating users choosing one answer by default when they are unsure about answer.

This specific simulation is using two uncorrelated skills and two answers. They are distributed in a way that there is same amount of each combination (1/4 of items). From previous work we know that uncorrelated skills will form distinct clusters. We included them in this simulation to better illustrate conditions of real data - as such clusters of levels exist there.

Projection of data simulated in this way has same properties as our real data. There are clusters (in this case caused by uncorrelated skills) which have items divided by correct answer.

We succseeded in simualting data with clusters of same answers and showing that this is case in our real data. Hovewer there is still one more thing to explain - clusters of uqesrions with same levels. Folloving same approach we can try to filter out users with large difference in how successfuk they are with solving different levels.

Weused histogram of differences (variance of solved levels) to choose some value for filtering of users. As more than half of users solves only one level median of differences is 0.0. So we choose one value just by looking at histogram. We can see in following figure that this dosnt really matter as filtering users in this way does not help with eliminating clustrs of same levels.

3.5 User similarity

For explaining patterns of aquestions from same level we have to dig deeper and understand different fpgroups of users. One particular way how to acjive this is using owrkflow similar to previous. Althought there will be one difference, we will be using similarity of users instead of items. (item-item similarity matrix, user-user similarity matrix). This also meqns that we have to transpose performance matrix - so each column represents one user. Calculating correlarion between all columns gives us user similarity matrix. There is no difference in projection step, only used matrix is interexchnaged.

W pe ca.n try ploting this result direcly using PCA. Hovewer we would notice that resulting image doesnt give us much information about users. reason for this is that two principal components reflect some property of data that we know about and dont wqnt to display. In particular users solving only one group are somewhat special. Columns representing users like this cannot be compared when they solved different levels - there is missinf information about their correlarion. PCA chooses this information with highest magnitude In presented figure this coresponds to choosen components reflecting solving of first and third level. (third principal domponent is corelates with solving of second level)

As users who solve only one level cause this behaviour. And they dont give us information about similarity of items in different levels we choose to exclude this users. After doing so we end up with projection similar to figure xx.

There are two visible clusters of users. Futher analysis shows that difference beween groups of users is caused by ?

3.6 Another contexts

4 Conclusion

We can conclude some recommendations. While trying to explain patterns in our particular used data we run into multiple situatutions hwich can repeat even when explaing some different data using similalr techniques.

Technique we used for calculating projections is quite common in area of Adaptife learning and recommender systems. CITE.

Following section will summarize recommendations useful for explaining results

it is useful to look at both item and user similarity - some patterns in data can be results of habbits of users. - we are using unsupervsed learning like techniques - we obtain some results but it is hard to explain why they are like they are.. one possibility how to explain results is using simulations - we explained one possible way of simulating performance matrix - data fome from users so identifing groups of users who behave same may be useful - we rhink most common groups which can be vosibke in data coming from online tutoring system are eager users and trolls - Main grohp of users tend to answer to quesrions in some regullar patrern based on their knowledge while trolls follow different pattern (for exampla use same answer on all questions) - looking at total similarity of items - typicaly correlates with first dimension of PCA - variance of performance entries of items

4.1 Limitations

- only one exercise - only performane was used for computing similarity of items, other sources of data were ignored .. as we thought there results here are most unexpected .. hovewer there may arise other problems when using different sourcess of data - we used only correctness of answers and ignore response time. It make sense in studied tutoring system but may be important in other systems.

A An appendix

Here you can insert the appendices of your thesis.