Masaryk University
Faculty of Informatics

Similarity of programming problems

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 have large amount of items which enables them to provide 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 problem of measuring similarity of programming problems. It explains difference between program and programming problem, which data we have available and techniques used for measuring similarity of problems. Second chapter advances level deeper and describe everything what is specific to data we used. First part of chapter describes programming environment of Robotanik and data from it. Second part focuses in detail on metrics we used in experiments. Last chapter gives overview of implementation and usage of metrics and their evaluation.

1 Similarity

In this chapter we will talk in general about questions in learning systems, and computing their similarity. Most of the chapter focuses on explaining what kinds of data are available when comparing questions in learning systems and techniques to do so. Last section describes goals of the thesis.

A lot of research has been dedicated to similarity in many different fields computer science like bioinformatics (sequence alignment, similarity matrix of proteins), information retrieval (document similarity), plagiarism detection and many more.

One closely related area is recommender systems which differs from problem similarity only slightly. Both areas are distinguishing users and items. Only difference is that we know how well user did while solving specific item and recommender systems use rating of the items.

Main difference is that we can use more data about problem. We also have some problem statement and data about performance of students when solving problem.

1.1 Items

In this work we use the term "items" (problems, questions, assignments) when we refer to single entry in educational system which users can answer to. Since many aspects of this work are generally applicable we decided to use this general term. In some learning systems this can refer to simple choice from two options in another complex tasks which user solves in matter of minutes. On other side of the spectrum are systems for teaching introductional programming. Users tend to spend few minutes solving each task and there is fewer of them.

To further specify the context of our research, we will describe characteristics of items. For computing similarity of items it is most important knowing which data are available to us. Therefore we describe items by sources of data can be used for measuring similarity.

- **Item statement:** specification of the item that a learner should solve, e.g., as a natural language description of the task.
- **Item solutions:** details about solutions obtained from learners or sample solution to item.
- **Learner's performance:** for example item solving times, correctness of answer, number of attempts needed.

This description of item is broad enough to cover most of learning systems. In next chapters we will discuss two systems in particular - umimecesky a umimematiku.

1.2 Why is similarity of items useful

As we mentioned previously key part of learning solving of educational items.

1.3 Computing similarity of items

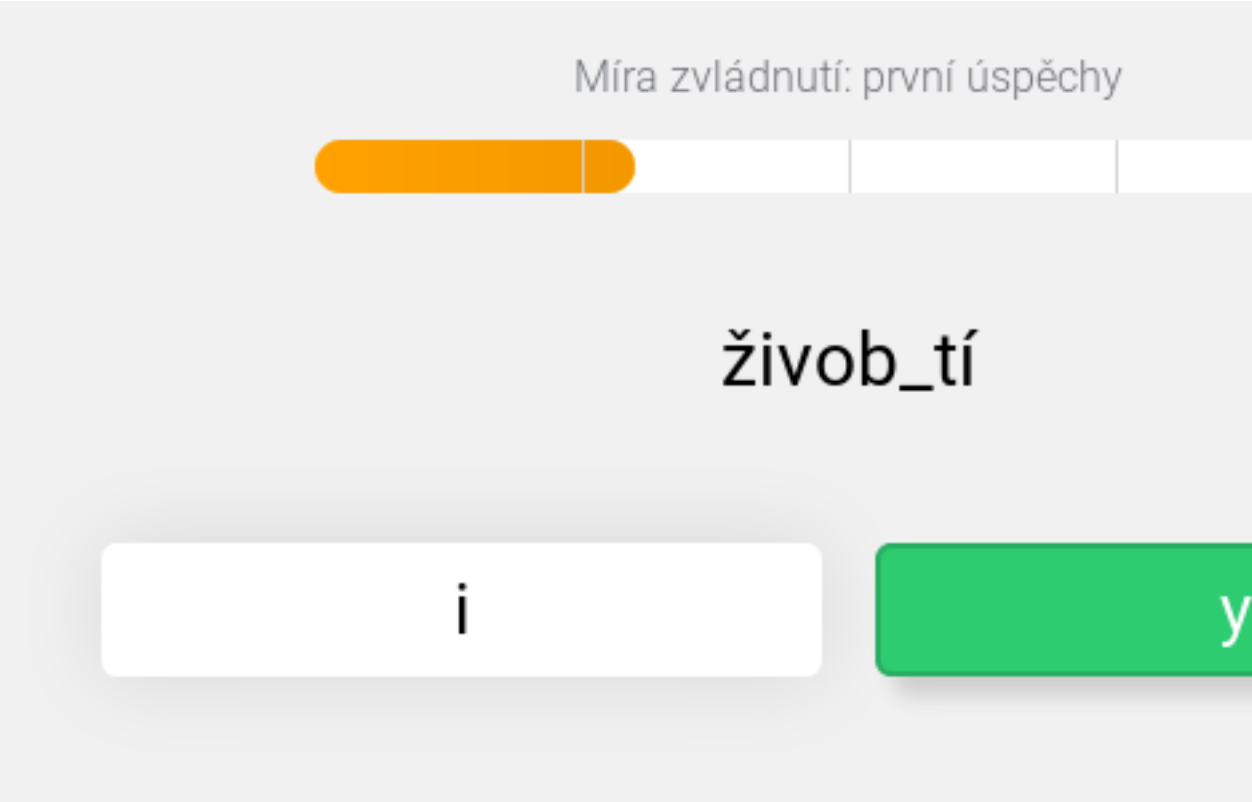
The general approach to measuring and using similarity of educational items

1.4 Used datasets

In our analysis we use both real data from educational system and simulated data. There is a reason why use both as only real-world data are useful for concluding any results. However evaluation of this data is often complicated as we do not know truth about many of their aspects. That's why we used simulated data for validating some of our conclusions. We will talk more in depth about how we generated simulated data in next chapter when describing their specific usage.

Most of used real-world data comes from system Umíme česky[1]. Later we have validated our results by also using data from its sibling

1. <https://umimecesky.cz/>



system Umíme matiku[2]. We think it is useful as data come from another context but are provided in same format and therefore can be used directly in previously created tools.

1.4.1 Umíme česky

Umíme česky[3] is system for practice of Czech grammar. System contains multiple exercise types, but in our analysis we use only one exercise - simple "fill-in-the-blank" with two possible answers. This type of exercise can be then viewed by student in multiple ways.

We focused only on "fill-in-the-blank" exercises but they can still be used to train many concepts of Czech grammar.

2. <https://umimematiku.cz/>
3. <https://umimecesky.cz/>

1.4.2 Umíme matiku

2 Evaulation

it is hard to say anything about data when tehre is a lot of it we especialy focused on systems which consist of TODO1000s of solvable items and even more users in cases like this it is not possible to look at data about each item 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 2.1 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.

2.1 Visible Properties of projection

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 using performance as data source this is especially items using same skill are projected near each other. E.g. figure 2.2 shows projection of questions about Czech language with highlighted a few similar words. This group consists of words witch are based on single word with added suffix.

After looking at image you can notice some regularities.

2.1.1 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.

1. First step we have to do is converting raw logged information about user answers to performance matrix. This matrix consist

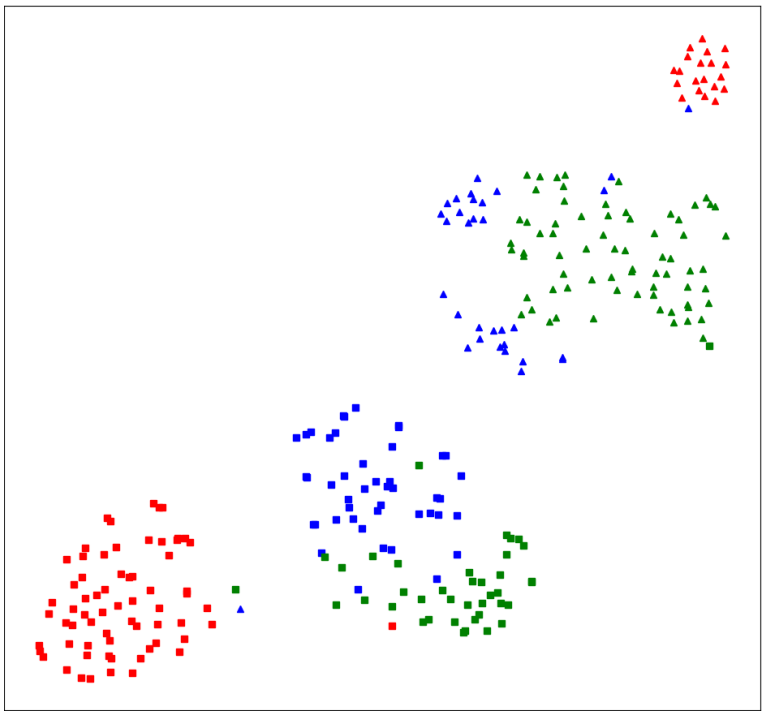


Figure 2.1: Basic projection of one knowledge component in Umíme česky

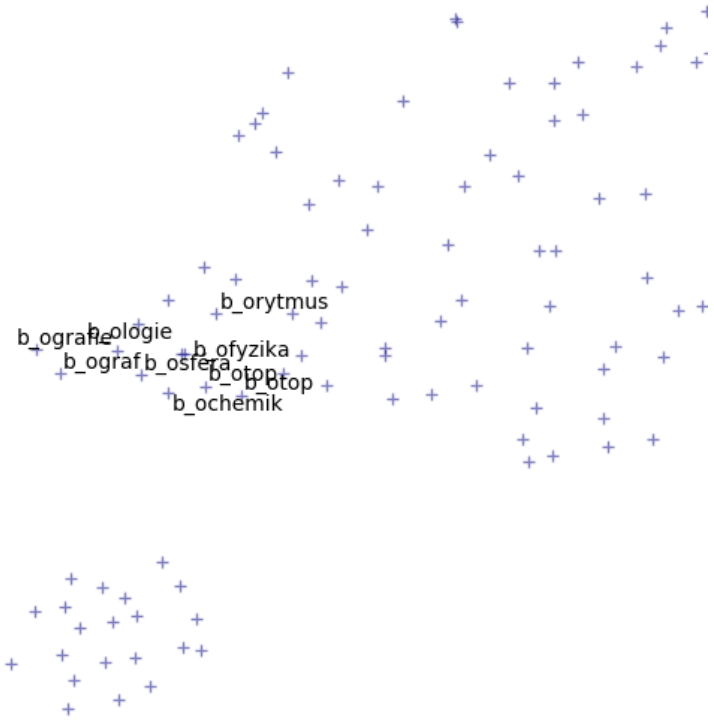


Figure 2.2: Similar words close to each other

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. TODO possible choices are described in article [] Performance matrix is relatively sparse as it is not common for users to solve all the items in the system.

2. Next step is computing similarity matrix. Matrix S is square matrix where each position S_{ij} denotes similarity of items i and j. In our case similarity is computed as correlation of two columns i and j in performance matrix. This means we look only at users who solved both items i and j and compute Pearson correlation coefficient of their performance.
3. The last (optional) step is producing 2D projection. This can be achieved by using many techniques for computing low dimensional projections.

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. 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 one less variation to results caused by algorithm. It is easier to compare results when altering metrics used fro 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.

2.2 Level regularity

When you look back at figure 2.1 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. As you can see each color (level) forms a distinct cluster.

As we mentioned before, only data about user performance 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 component "".

2.2.1 Sparseness of performance matrix

For purpose of determining this, we designed an experiment. writing script that simulated answers provided by virtual students. Main problem we wanted to recreate was missing data in performance matrix. Each simulated user was assigned to one or more levels. Simulated user solved one level and then with some probability continued to another.

Good enough as users do not solve levels in order from easy to hard. There are also 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. But this is not our case. The only case where projection is divided into clusters is when there is absolutely no performance data between any question of groups.

2.3 Different performance metrics

2.4 Answer regularity

2.5 Another context

3 Conclusion

A An appendix

Here you can insert the appendices of your thesis.