Automated Classification and Categorization of Mathematical Knowledge

Radim Řehůřek, Petr Sojka

Masaryk University, Faculty of Informatics, Brno, Czech Republic
xrehurek@fi.muni.cz, sojka@fi.muni.cz

Abstract. There is a common Mathematics Subject Classification (MSC) System used for categorizing mathematical papers and knowledge. We present results of machine learning of the MSC on full texts of papers in the mathematical digital libraries DML-CZ and NUMDAM. The F_1-measure achieved on classification task of top-level MSC categories exceeds 89%. We describe and evaluate our methods for measuring the similarity of papers in the digital library based on paper full texts.

1 Introduction

> We thrive in information-thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, *categorize*, catalog, *classify*, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats. (Edward R. Tufte)

Mathematicians are used to classifying their papers. One of the first mathematical classification schemes appeared in the subject index for *Pure Mathematics* of 19 volumes of the *Catalogue of Scientific Papers 1800–1900* [1]. This attempt was continued but not completed by the *International Catalogue of Scientific Literature (1901–1914)*. About two hundred classes were used. Headings in the *Jahrbuch* [2] may be considered as another classification scheme.

The Library of Congress classification system has 939 subheadings under the heading QA–Mathematics. Another schemes used in many libraries around the world are the Dewey Decimal system and the Referativnyi Zhurnal System used in the Soviet Union. To add to this wide variety of schemes, we may mention systems used by NSF Mathematics Programs, by various encyclopaedia projects such as Wikipedia, or by the arXiv Preprint project. However, the most commonly used classification system today is the Mathematics Subject Classification (MSC) scheme (http://www.ams.org/msc/), developed and supported jointly by reviewing databases *Zentralblatt MATH* (Zbl) and *Mathematical Reviews* (MR).

In order to classify papers some system of paper categorization has to be chosen. We may pick up some established system developed by human experts or we may try to induce one from digital library of papers by clever document clustering and machine learning techniques.

1.1 Mathematics Subject Classification

It is clear that no fixed classification scheme can survive longer time period, since new areas of mathematics appear every year. Mathematicians entered the new millennium with the MSC version 2000, migrating from MSC of 1991. Draft version of MSC 2010 has already been prepared and published at `msc2010.org` recently. The primary and secondary keys of MSC 2000, requested today by most mathematical journals are used for indexing and categorizing a vast amount of new papers—see the exponential growth of publications in Figure 1.

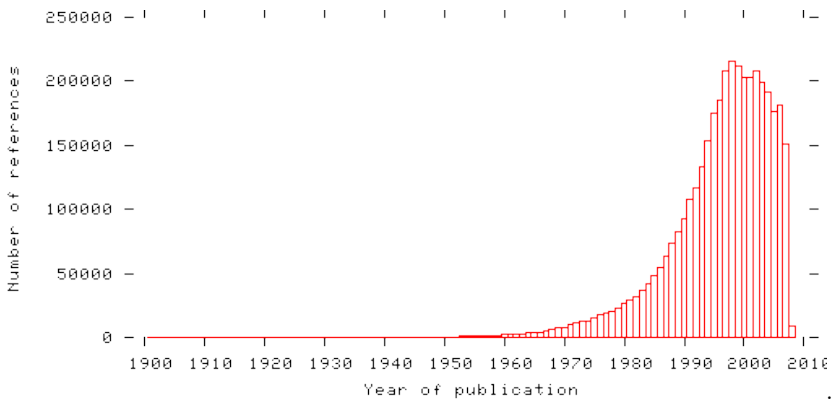

Fig. 1. Number of references in The Collection of Computer Science bibliographies (`http://liinwww.ira.uka.de/bibliography/`) as of March 2008

We believe that automated classification system, good article similarity measures and robust math paper classifiers allowing more focused math searching capabilities will help to tackle the future information explosion as predicted by Stephen Hawking:

> "If [in 2600] you stacked all the new books being published next to each other, you would have to move at ninety miles an hour just to keep up with the end of the line. Of course, by 2600 new artistic and scientific work will come in electronic forms, rather than as physical books and paper. Nevertheless, if the exponential growth continued, there would be ten papers a second in my kind of theoretical physics, and no time to read them."

Editors of mathematical journals usually require the authors themselves to include the MSC codes in manuscripts submitted for publication. However, most retrodigitized papers published before the adoption of MSC are not classified yet. Some projects, e.g. JAHRBUCH, use MSC 2000 even for the retroclassification of papers. Human classification needs significant resources of qualified mathematicians and reviewers. A similar situation is in the other retrodigitization projects such as NUMDAM [3] (`http://www.numdam.org`), or DML-CZ [4,5] (`http://www.dml.cz`): classifying digitized papers with MSC 2000 manually is expensive.

As there are already many papers properly classified (by authors and reviewers) in recent publications, methods of machine learning may be used to train an automated classifier based on the full texts author- and/or reviewer-classified papers.

This paper is organized as follows. In Section 2 we start by describing our data sources and preprocessing needed for our experiments. Section 3 on page 5 discusses results of automated document classification. In Section 4 on page 8 we show what results we obtained while computing mathematical document similarity. Finally, we discuss possible future experiments and work in Section 5 on page 13.

2 Data Preprocessing

> We run carelessly to the precipice, after we have put something before us to prevent us seeing it. (Blaise Pascal)

The data available for experiments are metadata and full texts of mathematical journals covered by the DML-CZ and NUMDAM projects.

2.1 Primary Data

During the first three years of the DML-CZ project, we digitized and collected data in the digital library, accessible via a web tool called Metadata editor (`editor.dml.cz`). To date (March 2008), in the digitized part there are 369 volumes of 14 journals and book collections: 1,493 issues, 11,742 articles on 177,615 pages.

From NUMDAM, we got another 15,767 full texts of articles (in simple XML format) for our research. We converted them into DML-CZ format as utf8 encoded text and excluded 134 articles due to inconsistencies such as having the same ID for parts of paper, invalid MSC etc. There were 5,697 papers tagged as English, 4,587 as French, 384 as Italian, 84 as German and there was no language tag for the remaining 4,881 papers available—language can be reliably detected by established statistical methods [6].

For experiments, we have used two types of data:

1. Texts from scanned pages of digitized journals (usually before 1990, where no electronic data are available). There are of course errors in full text, especially in mathematical formulae, as these were not recognized by OCR.

2. Texts from 'digital-born' papers, written in TEX, as papers of the journal *Archivum Mathematicum* (`http://www.emis.de/journals/AM/`) from years 1992–2007, where we had access to TEX source files. The workflow of the paper publishing process in some journals was modified somewhat so that all fine-grained metadata including the full text are exported for the digital library for long-term storage (CEDRAM project).

We started our experiments with retrodigitized articles, where texts were obtained by the OCR process [7].

After excluding papers with no MSC code we were left with 21,431 papers. From those, we only used papers tagged as English and with only a primary MSC classification (no secondary MSC) for our current experiments. This left us with 5,040 articles.

We started with our experiments with the task of classification of top-level (the first 2 digits) MSC categories. To ensure meaningful results, we used only a part of the text corpus: only top-level categories with more than 30/40/60 papers in them were considered. Without this pruning step, we could not expect the automated classifiers to learn well: given tiny classes comprising only a few papers, generalizing well is not straightforward. In this way, we were left with 31, 27 and 20 top-level MSC classes for the minimum 30, 40 and 60 papers per class limit, respectively. The total amount of articles after this pruning step is 4,618, 4,481 and 4,127 articles, respectively.

2.2 Preprocessing and Methods Used

It is widely known that the design of the learning architecture is very important, as is preprocessing, learning methods and their parameters [8].

For the purpose of building an automated MSC classification system, we chose the standard Vector Space Model (VSM) together with statistical Machine Learning (ML) methods. In order to convert the text in the natural language to vectors of features, several preprocessing steps must be taken—for a more thorough explanation, see e.g. [8]. A detailed description of all ML methods and IR notions is beyond the scope of this paper; the reader is referred to the overviews [9,10,11] for exact definitions and notation used.

The setup of the experiments is such that we run a vast array of training attempts in multidimensional learning space of tokenizers, feature selectors, term weighting types, classifiers and learning methods' parameters:

tokenization and lemmatization: the first part of the preprocessing relates to how the text is split into tokens (words)—alphabetic, lowercase, Krovetz stemmer [12], lemmatization, bi-gram tokenization (collocations chosen by MI-score);

feature selectors: how to choose the tokens that discriminate best—χ^2, mutual information (MI-score) [13,14,15];

feature amount: how many features are needed to classify best—500, 2,000 or 20,000 features [14];

term weighting: how the features will be weighted (*tfidf* variants [16] or [11, Fig. 6.15]) and smart weights normalizations (*atc* (augmented term frequency), *bnn* and *nnn*) [17];

classifiers: Naïve Bayes (NB), k-Nearest Neighbours (kNN), Support Vector Machines (SVM), decision trees, Artificial Neural Nets (ANN), K-star algorithm, Hyperpipes;

threshold estimators: how to choose the category status of the classifier based on a threshold—*fixed* or *s-cut* strategy for threshold setting [18];

evaluation and confidence estimation: how results are measured and how the confidence is estimated in them—Receiver Operating Characteristic (ROC), Normalized Cross Entropy (NCE) [19].

To give an example, evaluating one particular combination might mean that we tokenize the corpus using an alphabetic tokenizer, convert the tokens to lower case, select the best 2,000 tokens (words aka features aka terms) using χ^2 and weigh them using an *atc* scheme. One part of the corpus is then used for training the binary classifiers and the rest is evaluated to see whether the predicted MSC equals the expected MSC. Each binary classifier is responsible for one category (MSC class), and given a full text on input, returns whether the input belongs to the category or not. Each article may thus be predicted to belong to any number of categories, including none or all.

Out of the seven classifiers listed above, only the first three were used in the final experiments. The other four were discarded on the ground of poor performance in preliminary experiments not reported here. On the other hand, there are several recent hierarchical classification algorithms [20] that we did not have time to explore yet.

In order to evaluate the quality of each learned classifier, we compute an average of ten cross-validation runs. We measure micro/macro F_1, accuracy, precision, recall, correlation coefficient, break-even point and their standard deviations [11,8]. Since the popular accuracy measure is highly unsuitable for our task (extremely unbalanced ratio of positive/negative test examples), we will report results using the even more popular F_1 measure in this paper.

All these results are then compared to see which 'points' in the parameter space perform best. Our framework allows easy comparison of the evaluated parameters with visualization of the whole result space methods chosen—see multidimensional data visualization on Figure 2.

As the number of different learning setup combinations grows exponentially, methods that performed poorly in preliminary tests were excluded in the full testing.

3 MSC 2000 Automated Classification

> The classification here adopted has been the subject of more or less unfavourable criticism; the principal objection to it, however, seems to be that it is different from any of those previously employed, and is therefore to this extent inconvenient without any obvious advantage in the innovations. (J. A. Allen, [21])

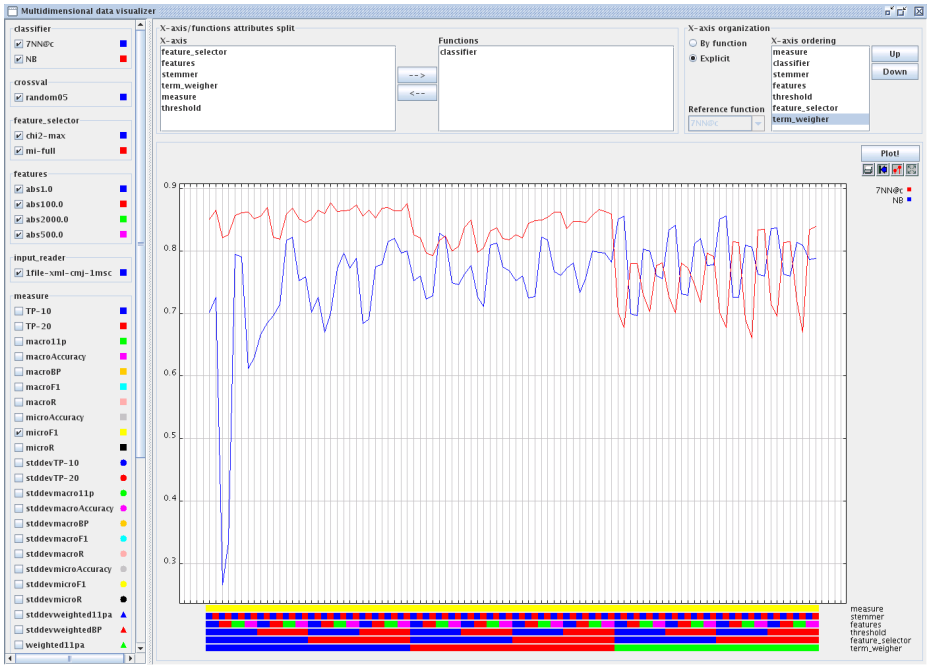

Fig. 2. Framework for comparing learning methods [8]. The two differently colored curves correspond to the chosen learning methods (k-NN, Naïve Bayes in the legend on the right). From the colors below chosen function values, one immediately sees which combination (at the bottom) of preprocessing methods leads to which particular value.

A detailed evaluation of classification accuracy shows that, while automatically classifying the first two letters of primary MSC, we can easily reach an 80% F_1 classification score with almost any combination of methods. With fine-tuning the best method (Support Vector Machines with a large number of features seems to be the winner) we can increase the F_1 score to 89% or more. The micro-averaged accuracy measure is above 99%, but is uninteresting as the baseline score, which can be achieved by a trivial reject-all classifier, is as high as $30/31 = 97\%$. The same difficulty does not arise with microaveraged F_1, where trivial classifiers score under 6%. In this light, our best result of nearly 90% F_1 score is quite encouraging.

In Figure 3 on the facing page there is a side-by-side plot of three different corpora which result from setting the 30/40/60 minimum articles per category threshold. The intuition is that, given less training examples, the task of learning a classifier would become harder and classification accuracy would drop. This can indeed be observed here. On the other hand, the drops are not dramatic but rather graceful (about two F_1 percentage points for going from a minimum of 60 to 40, and another 1% for going from 40 to 30). Also to be noted is another

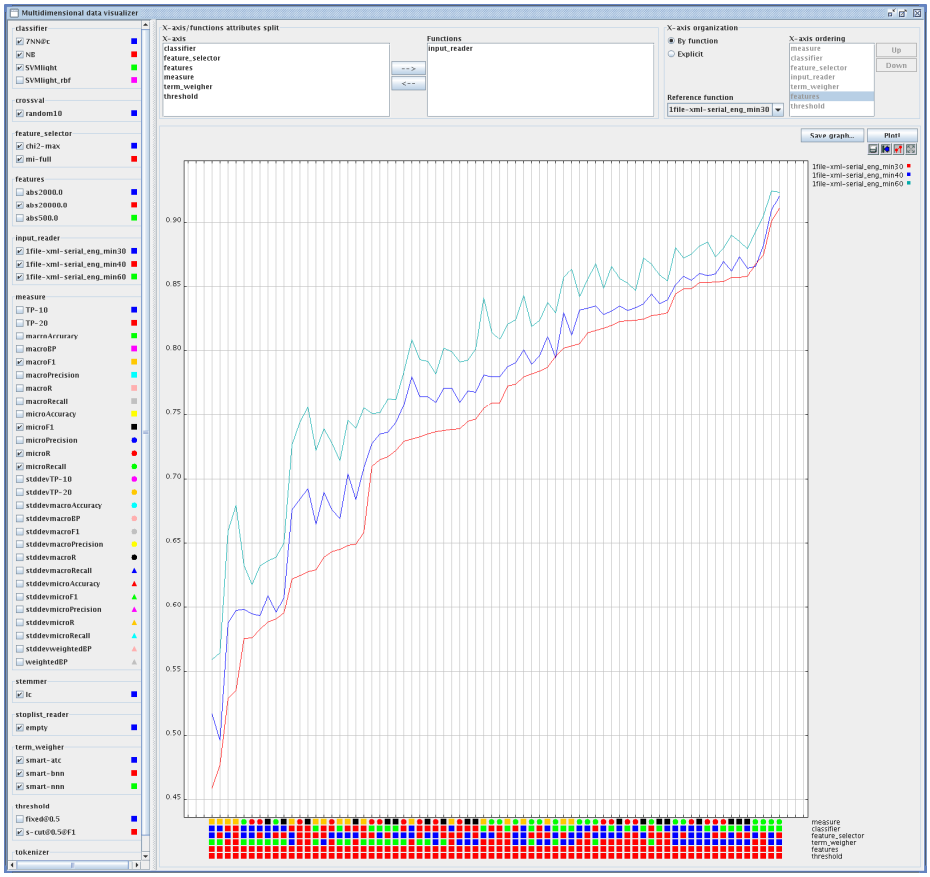

Fig. 3. Dependency of performance on the number of examples per class limit. From the three curves one can see that by increasing the threshold of minimum category size one gets better results in every aspect (color square combination at the bottom).

factor contributing to this drop—with a lower article per category threshold, we are in fact classifying into more classes (recall that the number of classes for the 60, 40 and 30 threshold is 20, 27 and 31 classes, respectively). Again, this makes more room for error and lowers the score.

Figure 4 on the next page enables us to examine the best performing combinations of methods and parameters. It may be observed that the best classifiers are exclusively SVM and kNN; the performance of NB depends heavily on term weighting. Also the aggressive feature selection of only 500 features performed poorly. The best result of the micro-averaged F_1 score of 89.03% was achieved with SVM with linear kernel, χ^2 feature selection of 20,000 features, *atc* term weighting and decision threshold selected dynamically by *s-cut*. In the light of

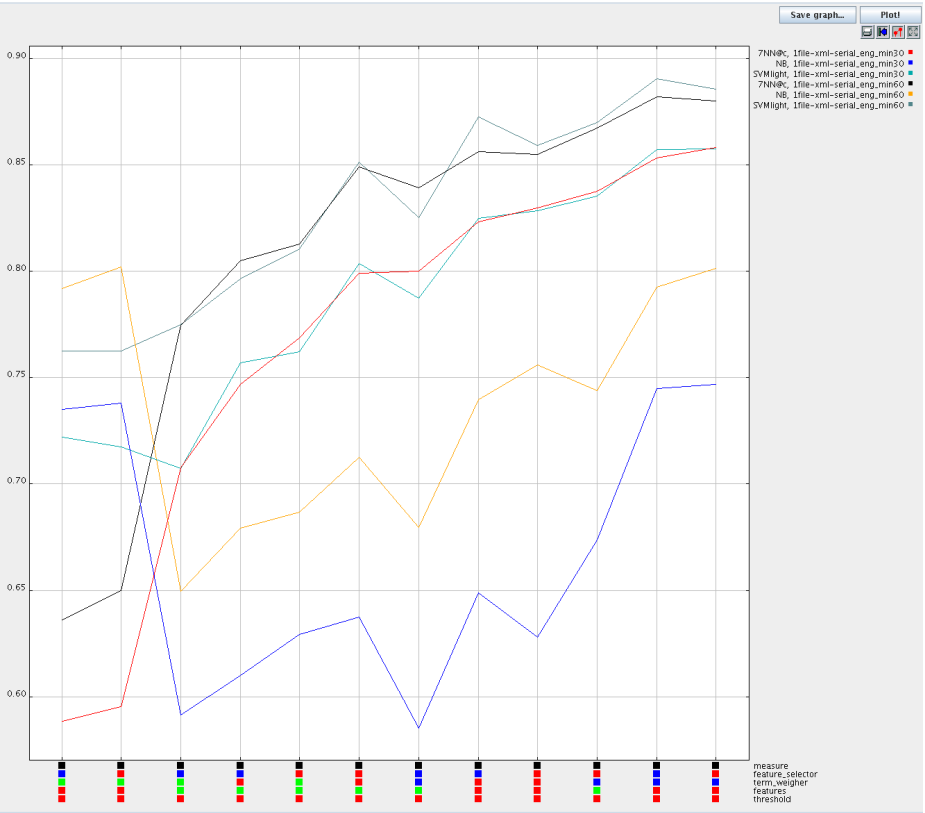

Fig. 4. Classifiers' learning methods comparison by F_1 measure. SVM and kNN run hand in hand while NB lags behind. The major influence is due to the threshold on minimum category size (see Figure 3 on the preceding page).

the previous comment, it is unsurprising that this maximum occurred in the dataset selected with a minimum article per category threshold of 60. F_1 scores at the very same configuration, but with a threshold of 40 and 30 articles per category read 86.28% and 85.72%, respectively.

Similarly, we measure and can visualize training times (computation expense) for every method tried. Many of these are computationally expensive—it takes days to weeks on a server with four multithreaded processors to compute all the results to visualize and analyze.

4 Mathematical Document Similarity

It's false to assume that mathematics consists of discrete subfields, it's false to assume that there is an objective way to gather those subfields into main divisions,

> and it's false to assume that there is an accurate two-dimensional positioning of the parts. (Dave Rusin [22])

Recall that one of the purposes of the automated MSC classification detailed above is to enable a similarity search. Given MSC categories, the user may browse articles with similar MSCs and thus (hopefully) with similarly relevant content.

But we have also been intrigued by similarity searches based on raw full text, and not on metadata such as MSC codes. This differs in that there is no predefined class taxonomy that the articles ought to follow (such as MSC). The similarity of two articles is gauged directly based on the articles' content, with no reference to human-entered or human-revised metadata.

Because fine linguistic analysis tools would be ineffective (recall that our texts come from OCR, with errors appearing as early as at the character level), we opted for 'a brute' Information Retrieval approach. Namely, we tried computing paper similarities using *tfidf* [16] and Latent Semantic Analysis (LSA) [23] methods. Again, both use a Vector Space Model, first converting articles to vectors and then using the cosine of the angle between the two document vectors to assess their similarity. [11] The difference between them is that while *tfidf* works directly over tokens, LSA first extracts concepts, then projects the vectors into this conceptual space where it only computes similarity. For LSA we chose the 200 top latent dimensions (concepts) to represent the vectors, in accordance with standard Information Retrieval practise [23].

Evaluating the effectiveness of our similarity schemes is not as straightforward as in the classification task. This is due to the fact that, as far as we know, there exists no corpus with an explicitly evaluated similarity between each pair of papers. In this way, we are left with two options: either constructing such corpus ourselves, or approximating it. As the first option appears too costly, we decided to assume that MSC equality implies content similarity. Accordingly, we evaluated how closely the computed similarity between two papers corresponds to the similarity implied by them sharing the same MSC.

Again, to avoid data sparseness, we only took note of the top MSC categories (first two letters of the MSC codes). In Figures 5 and 6 on page 11 there are *tfidf* and LSA plots of similarities between all English papers in our database that are tagged only with a primary MSC code.

Two things can be seen immediately from the plot:

- articles within one top MSC group are usually very similar (lighter squares along the diagonal);
- the similarities of articles from different MSC groups are low (dark rectangles off diagonal).

There are also exceptions, such as patches of light colour off the diagonal as well as dark patches within the MSC group squares. This is however to be expected from noisy real-world data and cannot be fixed nor explained without actually inspecting the articles by hand. Clear small square areas in matrix detail on Figure 7 on page 12 show that papers exhibit similarity of MSC even when sharing MSC code prefix of length 3 or higher.

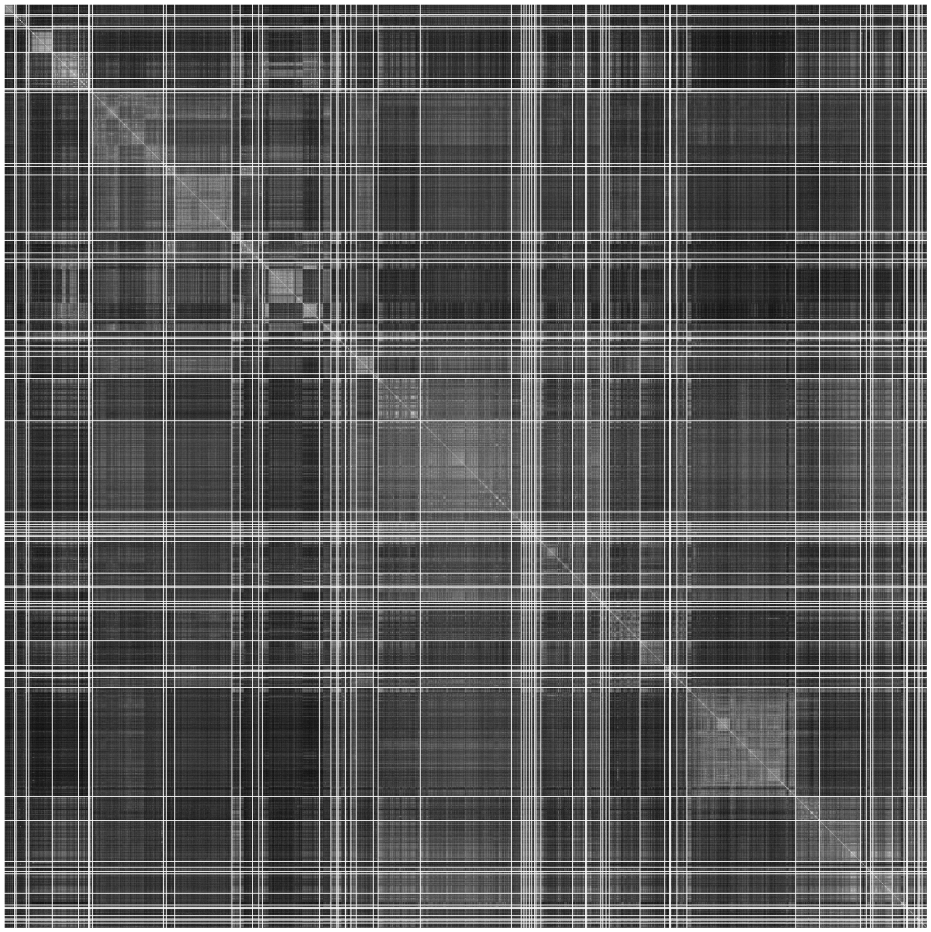

Fig. 5. MSC-sorted documents' similarity matrix computed by *tfidf*. The axes of this 5104×5104 matrix are articles, grouped together by their MSC code (white vertical and horizontal lines separate different top-level MSC categories) and sorted lexicographically by full five letter MSC code. The intensity of the plot shows similarity, with white being the most similar and black being completely dissimilar. Note that because the ordering of articles along both axes is identical, all diagonal elements must necessarily be white (completely similar), as each article is always fully similar to itself.

4.1 Experiments with Latent Semantic Analysis

Next experiment we tried with Latent Semantic Analysis [23] was to see which concepts are the most relevant ones.

There were papers in several different languages in the *Czechoslovak Mathematical Journal* (CMJ). When we listed the top concepts in LSA of the CMJ

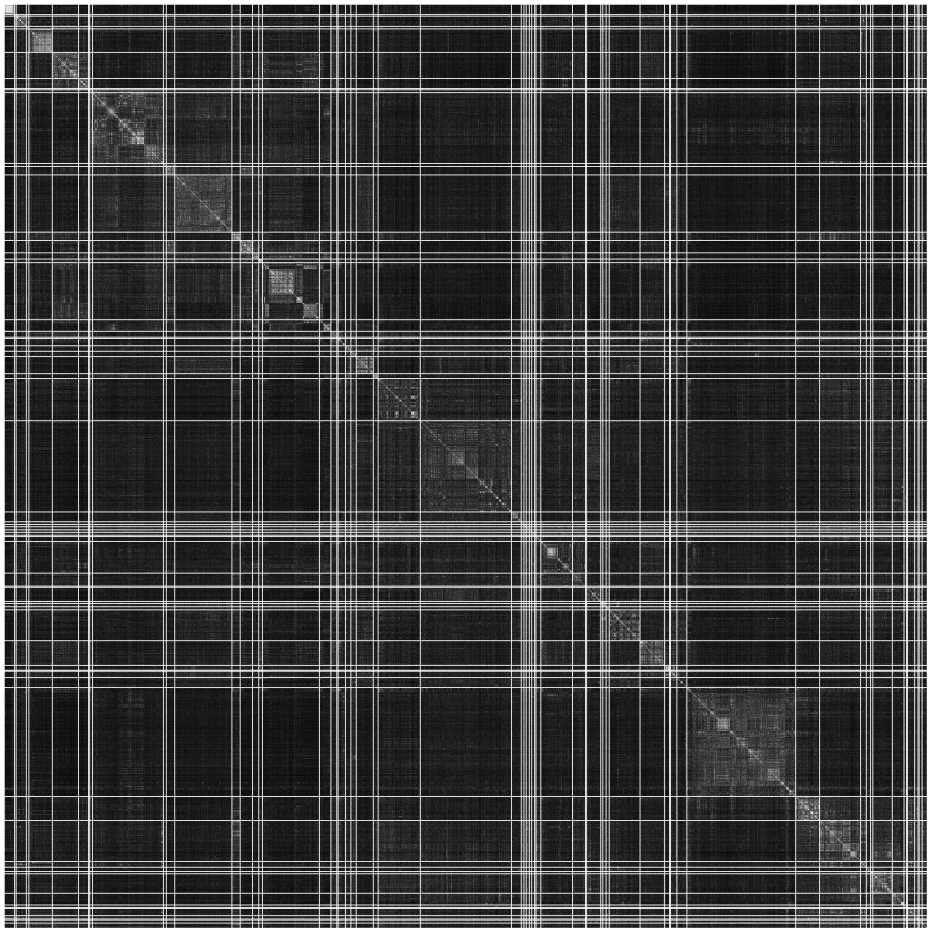

Fig. 6. MSC-sorted documents' similarity matrix computed by LSA. Interpretation is identical to Figure 5.

corpus, it was clear that the first thing the method was going to decide was its language, as the first terms of top concepts are:

1. 0.3 "the" +0.19 "and" +0.19 "is" +0.18 "that" +0.15 "of" +0.14 "we" +0.14 "for" +0.11 "ε" +0.11 "let" +0.11 "then" +...
2. −0.41 "ist" −0.40 "die" −0.28 "und" −0.26 "der" −0.23 "wir" −0.21 "für" −0.17 "eine" −0.17 "von" −0.14 "mit" −0.13 "dann" +...
3. −0.31 "de" −0.30 "est" −0.29 "que" −0.27 "la" −0.26 "les" −0.2 "une" −0.2 "pour" −0.20 "et" −0.18 "dans" −0.18 "nous" +...
4. −0.36 "что" −0.29 "для" −0.23 "пусть" −0.19 "из" −0.19 "если" −0.16 "так" −0.16 "то" −0.14 "на" −0.14 "тогда" −0.131169 "мы" +...
5. −0.33 "semigroup" −0.25 "ideal" −0.19 "group" −0.18 "lattice" +0.18 "solution" +0.16 "equation" −0.16 "ordered" −0.15 "ideals" −0.15 "semigroups" +...

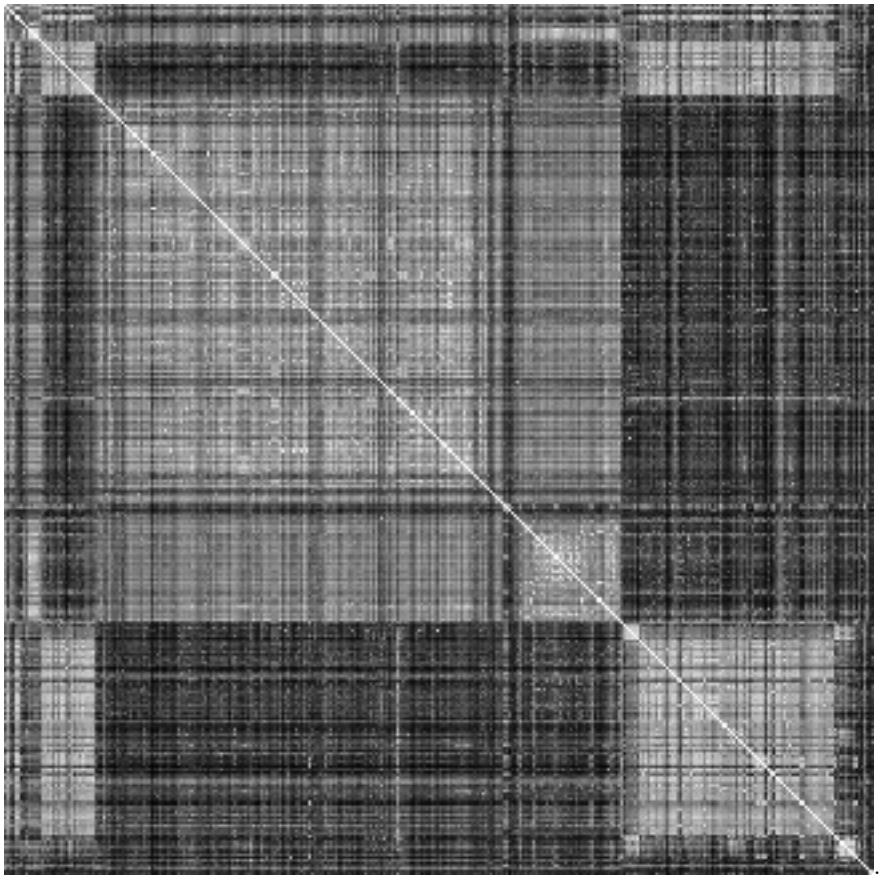

Fig. 7. Detail of MSC-sorted documents' similarity matrix computed by LSA for top-level MSC code 20-xx *Group theory and generalizations*. The white lower right square corresponds to the 20Mxx *Semigroups* subject papers. We can see strong similarity of 20Mxx to 20.92 *Semigroups, general theory* and 20.93 *Semigroups, structure and classification* (white lower left and upper right rectangles).

6. 0.46 "graph" +0.40 "vertices" +0.36 "vertex" +0.23 "graphs" +0.2 "edge" +0.19 "edges" −0.18 "ε" −0.15 "semigroup" −0.13 "ideal" +...
7. 0.81 "ε" −0.25 "semigroup" −0.16 "ideal" +0.12 "lattice" −0.11 "semigroups" +0.10 "i" −0.1 "ideals" +0.09 "ordered" +0.09 "ř" −0.08 "idempotent" +...
8. 0.29 "semigroup" −0.22 "space" +0.2 "ε" +0.19 "solution" +0.19 "ideal" +0.18 "equation" +0.16 "oscillatory" −0.15 "spaces" −0.16 "compact" +...

The first concepts clearly capture the language of the paper (EN, DE, FR, RU), and only then topical term-sets start to be grabbed. It is not surprising—the classifiers then have to be trained either for every language, or the document features have to be chosen language-independently by mapping words to some

common topic ontology. To the best of our knowledge, nothing like EuroWordNet for mathematical subject classification terms or mathematics exists.

Given the amount of training data—papers of given MSC code for given language—we face the sparsity problem for languages such as Czech, Italian, German and even French presented in the digital library.

When we trained LSA on the monolingual corpora of *Archivum Mathematicum*, where mathematics formulae were used during tokenization (subcorpus created from original TEX files), we saw that even in the first concepts, there was significant proportion of mathematical terms with high weights in concepts created by LSA:

1. −0.32 "t" −0.24 "ds" −0.17 "u" −0.17 "_" −0.17 "x" −0.15 "solution" −0.12 "equation" −0.11 "q" −0.11 "x_" −0.11 "oscillatory" +...
2. 0.28 "ds" +0.28 "t" −0.22 "bundle" −0.16 "natural" +0.15 "oscillatory" −0.15 "vector" +0.13 "solution" −0.13 "connection" −0.13 "manifold" +0.11 "t_0" +...
3. −0.22 "bundle" +0.19 "ring" −0.17 "natural" −0.16 "oscillatory" +0.15 "fuzzy" −0.15 "ds" +0.12 "ideal" −0.11 "t" −0.11 "r_0" −0.11 "nonoscillatory" +...

It supports the idea that mathematical formulae have to be taken into account—having robust math OCR and finding its good discriminative feature representation we may get much better similarity and classification results in the future.

5 Conclusions and Future Work

> Words differently arranged have a different meaning,
> and meanings differently arranged have different effects.
> (Blaise Pascal)

Our results convincingly demonstrated the feasibility of a machine learning approach to the classification of mathematical papers. Although we compared and reported the results according to the F_1 measure, our approach can easily be tweaked to favour a different trade-off between higher recall and/or precision. Results in the form of guessed MSC and similarity lists are going to be directly used in the DML-CZ project.

Given enough data, when we extrapolate the best results of preliminary experiments done on our limited data, with linear machine learning methods (creating separable convex spaces in multidimensional feature space) we were able to approach a very high precision of 96% and recall of 92.5%, which are the current bests, for a combined F_1 score of well over 90%. Future research thus extends to evaluating the classification on all 64 top MSC categories, and using hierarchical classifiers to cover the full MSC taxonomy. With ambitions for even higher recall, there are several approaches, namely to either improve the preprocessing for vectors representing the documents by NLP techniques (characteristic words, bi-words, etc.) or use higher order models (deep networks). Mainstream machine learning research was concentrated on using "convex", shallow methods (SVM,

shallow neural networks with back-propagation training) so far. State-of-the-art fine tuned methods allow very high accuracy even on large scale classification problems. However, the training of these methods is exceptionally high and the models are big. Using the ensembles of classifiers makes the situation even less satisfactory (size even bigger), and the final models need to be regularized. In future, we plan to try new algorithms for a hierarchical text classification [20] and training large models with non-convex optimization [24] that may give classifications that does not exhibit overfitting.

Further studies will encompass a fine-grained classification trained on bigger collections (using MSC tagged mathematical papers from (`ArXiv.org`), growing NUMDAM and DML-CZ libraries etc.), and a rigorous measure confidence evaluation [19].

For final large scale applications scaling issues, and fine-tuning the best performance by choosing the best set of preprocessing parameters and machine learning methods remains to be done. We will watch Apache Lucene Mahout project's code when scalability of machine learning will arise as a serious issue.

Acknowledgement This study has been partially supported by the grants 1ET208050513 and 1ET208050401 of the Academy of Sciences of the Czech Republic and 2C06009 and LC536 of MŠMT ČR.

References

1. Royal Society of London: Catalogue of scientific papers 1800–1900 (1908) Volumes 1–19 and Subject Index in 4 vols. published 1867–1925; free electronic version available by project Gallica `http://gallica.bnf.fr/`.
2. (Jahrbuch) In Ohrtmann, C., Müller, F., Eds.: Jahrbuch über die Fortschritte der Mathematik (1868–1942). Volume 1–68. Druck und Verlag von Georg Reimer, Berlin (1871–1942) electronic version available by project ERAM `http://www.emis.de/projects/JFM/`.
3. Bouche, T.: Towards a Digital Mathematics Library? In Rocha, E.M., ed.: CMDE 2006: Communicating Mathematics in the Digital Era. A.K. Peters, MA, USA (2008) 43–68.
4. Sojka, P.: From Scanned Image to Knowledge Sharing. In Tochtermann, K., Maurer, H., Eds.: Proceedings of I-KNOW '05: Fifth International Conference on Knowledge Management, Graz, Austria, Know-Center in coop. with Graz Uni, Joanneum Research and Springer Pub. Co. (2005) 664–672.
5. Bartošek, M., Lhoták, M., Rákosník, J., Sojka, P., Šárfy, M.: DML-CZ: The Objectives and the First Steps. In Borwein, J., Rocha, E.M., Rodrigues, J.F., Eds.: CMDE 2006: Communicating Mathematics in the Digital Era. A. K. Peters, MA, USA (2008) 69–79.
6. Dunning, T.: Statistical identification of language. Technical Report MCCS 94-273, New Mexico State University, Computing Research Lab (1994).
7. Sojka, P., Panák, R., Mudrák, T.: Optical Character Recognition of Mathematical Texts in the DML-CZ Project. Technical report, Masaryk University, Brno (2006) presented at CMDE 2006 conference in Aveiro, Portugal.

8. Pomikálek, J., Řehůřek, R.: The Influence of Preprocessing Parameters on Text Categorization. International Journal of Applied Science, Engineering and Technology **1** (2007) 430–434.
9. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47.
10. Yang, Y., Joachims, T.: Text categorization. Scholarpedia (2008) `http://www.scholarpedia.org/article/Text_categorization`.
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008).
12. Krovetz, R.: Viewing morphology as an inference process. In: Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Linguistic Analysis (1993) 191–202.
13. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In Fisher, D.H., ed.: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US, Morgan Kaufmann Publishers, San Francisco, US (1997) 412–420.
14. Galavotti, L., Sebastiani, F., Simi, M.: Experiments on the use of feature selection and negative evidence in automated text categorization. In Borbinha, J.L., Baker, T., Eds.: Research and Advanced Technology for Digital Libraries, 4th European Conference, ECDL 2000, Lisbon, Portugal, September 18–20, 2000, Proceedings. Volume 1923 of Lecture Notes in Computer Science., Springer (2000) 59–68.
15. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research **3** (2003) 1289–1305.
16. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management **24** (1988) 513–523.
17. Lee, J.H.: Analyses of multiple evidence combination. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Combination Techniques (1997) 267–276.
18. Yang, Y.: A Study on Thresholding Strategies for Text Categorization. In Croft, W.B., Harper, D.J., Kraft, D.H., Zobel, J., Eds.: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001), New York, ACM Press (2001) 137–145.
19. Gandrabur, S., Foster, G., Lapalme, G.: Confidence Estimation for NLP Applications. ACM Transactions on Speech and Language Processing **3** (2006) 1–29.
20. Esuli, A., Fagni, T., Sebastiani, F.: Boosting multi-label hierarchical text categorization. Information Retrieval **11** (2008).
21. Allen, J.A.: The international catalogue of scientific literature. The Auk **21** (1904) 494–501.
22. Rusin, D.: The Mathematical Atlas—A Gateway to Modern Mathematics (2002) `http://www.math-atlas.org/welcome.html`.
23. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407.
24. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In Schölkopf, B., Platt, J., Hoffman, T., Eds.: Advances in Neural Information Processing Systems 19. MIT Press, Cambridge, MA (2007) 153–160.