Proceedings of the 9th International Conference on Applied Informatics
Eger, Hungary, January 29–February 1, 2014. Vol. 1. pp. 19–26
doi: 10.14794/ICAI.9.2014.1.19

Email labelling by rough clustering*

László Aszalós, Mária Bakó, Tamás Mihálydeák

University of Debrecen
aszalos.laszlo@inf.unideb.hu, bakom@unideb.hu
mihalydeak.tamas@inf.unideb.hu

Abstract

Previously, there were little possibilities to sort mails and later emails: we could only arrange them into folders. One mail or email could be put into exactly one folder, based on sender, subject or priority. Later in Gmail the labelling of emails was introduced: virtual folders were generated by the multiple labels that could be assigned to one email. This kind of labeling was taken by other mailers, photo and music organizer softwares, too.

It is a pleasure to use a well organized collection, but usually paintaking to set up the its labelling. We simplify these kind of tasks by using our experience in rough set theory and clustering. The clustering is a well-known part of the data mining, where the elements are grouped by their similarity. The similarity is an inexact concept in real life, e. g. we easily mix up two Japanese persons, however a Chinese man easily differentiates them. We suggest that the rough clustering could help to combine data based on similarities from different sources, because it has some error-correcting property.

In this article we present a method to label emails and its mathematical background.

Keywords: rough set, correlation clustering, data mining

MSC: 03E02, 62H30, 90C27

1. Introduction

For thousand of years mankind only needed to store and organize physical objects for easy retrieving: papyrus rolls, books, documents, paintings, photos, jewellery, etc. Hence many organization methods have been created and used. The common property of all of them is that one object can be at only one place. If we want to reach it from different directions, we need to use some kind of pointers, e.g. card index.

*The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

In the computer era the situation have changed partly. Henceforward we use folders to store files, but using soft or hard links we can virtually put one file into several folders. Moreover it is trendy to use tags/labels to tag music files, pictures or articles of a blog or a journal, and use one tag or more to create virtual folders to retrieve some files or articles. The same is true for emails. Today only a few mailers do not allow to tag and retrieve mails by tags.

A well ordered collection is a treasure and joy to use. On the other hand, to regularly use a heap of objects without any order or structure is a big pain. To tell the truth, the ordering and structuring of a mass of objects is at least as painful. Maths could help in such cases. The clustering is a class of methods which discovers the structure of the objects, which can be treated as a preprocessing of the mass for the organization.

In this article we suggest to use a not so known method: rough clustering, to help the organization process. We present it by organizing emails.

The structure of the article is the following: in the next chapter we show a relatively fresh clustering method we used at our experiments. With this we sketch the main concepts of the rough set theory, and how it can combined with correlation clustering. After reviewing the theoretical background we switch over to a specific problem. To compare the emails we need to define their similarity or distance. We had chosen the first one, so we list the properties of emails we used at the comparisons. We touch upon the generation of a tolerance relation by this similarity measure. Next we show the results of the rough clustering at different parameters, and we suggest how these results can be used in the practice. Finally, we conclude our paper with a summary and outlook on.

2. Correlation clustering

The aim of clustering is to discover the structure and group the objects based on similarity, without any previous information of its structure. We would like similar objects to get into the same clusters, and different object to get into different groups. This condition is very general, so we cannot wonder how many different clustering methods exist. The most known methods are the hierarchical clusterings and the centroid methods, but there are many distribution models, density models, graph-based models, too. In practice, the crisp methods are the most common. Here, one object could belong at most one cluster. Recently, the soft methods are beginning to be more and more honoured, due to their practical applications.

Most of the classification methods are based on some distance methods. Based on this distance we can say that two objects are similar or dissimilar. The correlation clustering [3] is different. It uses a similarity relation, so two objects are similar if this relation holds, and if doesn't, then they are dissimilar. This relation is a tolerance relation, i.e. symmetric and reflexive, but not necessary transitive.

Although the original definition used a fully connected graph, we can easily generalize it. This means that the similarity relation is not defined for all pairs of objects, hence it is a partial relation.

The correlation clustering minimizes the disagreements: the number of pairs of dissimilar objects within clusters plus the number of pairs of similar objects in different clusters.

The fact, that the goodness of a clustering is measured by a number, enables us to compare the different clusterings/partitions. This comparability of clusterings is not common, at other methods thumb rules help the users to choose the right parameters: to get a good clustering.

But here is not the case. The comparison enables us to choose the best one. The result of a correlation clustering of some sets of objects is a partitioning where this sum is minimal. Unfortunately the number of partitioning is an exponential function of the number of objects, so at practical cases we can only approximate the optimal partitioning.

3. Rough clustering

By the definition of the correlation clustering it is possible that several equally good solutions exist. This enables us to construct a rough clustering. But at first we review the rough set theory.

In classical set theory the membership function is a bivalent condition: an element belongs to a set, or it does not. At fuzzy set theory the membership function has value in the real unit interval $[0, 1]$, and can take any element in this interval. At rough set theory a membership function with three values can be defined directly: an element is surely in a set, an element is surely not in a set and we have no/we cannot have enough information. Based on this membership function we can define the lower approximation (set of elements are surely inside) and the upper approximation (set of elements are possibly inside) of a set. The pair of lower and upper approximation defines a rough set [4, 5].

We remark, that in rough set theory the typical approach is different, at original state it starts from an equivalence relation, and uses it to define the lower and upper approximations [8, 9].

We use this approach, because we define rough clustering with object-based approximation [1]. Let's assume that we have several equally good (the result of the sums are the same) solution for correlation clustering. Then the *lower approximation* of an object is the intersection of the clusters of the partitions containing it. Similarly the *upper approximation* of an object is the union of the clusters of the partitions containing it. The idea behind this definition is the following: the most similar objects are those that we cannot separate in any way by clustering.

The rough set theory is based on sets, as sets are approximated with sets. Here we use the tools of the rough set theory to approximate objects with sets. It is obvious that an object is element of its lower approximation, and it can be easily checked that the lower approximation is subset of the upper approximation. Here, this union is approximated with objects which are in all sets of this union.

We recall that the correlation clustering produces the best partitions — where

the disagreement is minimal. Hence it describes well the structure of the elements. But it can happen that the neighbourhood of an object is such that there does no exit a unique best cluster in it. Two or maybe more equally good, but different clusters contain this object. What we can know, that by taking any of the best partitions, this object will be in the same cluster with the elements of its lower approximation. If object x is in the same cluster as object y at each optimal partition, then x is in lower approximation of y, and alike y is in lower approximation of x. The transitivity and reflexivity of this lower approximation can be proved easily, so based on lower approximation of objects we can define an equivalence relation on the objects, and we can define *base set* as the factor structure of the object based on this equivalence relation. We remark, that this approach is not the standard, usually the lower approximation is based on the equivalence relation, and not in reverse, as seen here.

The lower approximation — and hence the created base sets — is defined in such a way, that the objects in this approximation cannot be distinguished by the tolerance relation. So these objects are very similar. This is the reason why we suggest to transfer the tags automatically in Section 6.

The tests shows that the lower and upper approximations of the objects in general are not the same. The boundary — the difference of the upper and the lower approximation — of an object means a lower level of similarity, because there exists an optimal partition which differentiates the object in the boundary and the original object. Hence in Section 6 we suggest only a manual tag transfer, to add human control to it.

4. Similarity of emails

In order to apply the rough clustering introduced before we need a partial similarity relation on emails. We usually have an impression whether two emails are similar or not, and can decide this within seconds. But if we have thousands of emails then we need to decide about similarity millions-fold. Of course, we can leave this task to computers. They can do it, if they get proper conditions to examine.

When we compare emails, or when we organize them we use the address of the sender or receiver, the subject line, and the subject (theme) of the mail. Sometimes the date is important: at a recurring event which repeats yearly, there is a shallow similarity between fresh and two year old letters.

For testing we take the mails belonging to some project. This means 433 mails, some of them are sent, some of them are received by one of the authors. The mails incorporate three years and many subjects.

Hence for each mail we collected all the addresses appearing in their header, except for the author's. We can treat the similarity of two emails by addresses as a bivalent function: similar if there is a common name, and dissimilar otherwise; or we can use the Jaccard similarity coefficient: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$, where A and B are the sets of addresses in the emails.

Although many people can compose emails by pushing the reply button only, the same subject is a good indicator of similarity. Of course the string of subject needs to be cleaned from the sequence of "RE:" and "Fwd:".

The subject or subjects of the body of the text is another indicator of the similarity. Here we have the same problem as before: somebody instead of composing a new letter just replys to an old one, does not change the title, does not delete the quoted text, only adds one or several new lines. Meanwhile others delete most of the quoted text and insert their answers between the lines of the remaining quotations. In the first case we cannot take into consideration the quoted text, and the latter case we need to. By this, we just constructed a prototype. We did not check whether an email completely quotes some other mails or not. Our decision was to take into account the quoted text, too.

Most of the mails contain common texts: opening, closing and pre-closing formulas. They are mostly indifferent. Of course we could differentiate mails by their formal or informal closings, but many users have letter templates they use for any mails. Moreover we are not interested in the most common words: *the*, *of*, *an*, etc. In the text mining they are called *stop words* [7]. The list of stop words can be constructed by a computer, but our letters are written in Hungarian, and this is an agglutinative language, hence the algorithm would be very complicated. So we have used the old Unix tool [6] to construct the word frequency list. As we developed a prototype we filtered this list by hand to get the stop words list.

Next we took the body of each email, calculated the word frequency list of non stop words, and took the Jaccard coefficient of the n most frequent words.

Finally we took Jaccard coefficient of the attachments, and calculated the weighted sum of the different similarity relations.

This method gives a real number x for each pair of emails. We presented the matrix of these numbers as a picture on Fig 1. As the number x is higher and higher its pixel becomes darker.

From these numbers we need to create a tolerance relation. If the number x, which refers to the similarity is high, then we can treat two mails as similar. If this number is low, then the mails are dissimilar. If the number is medium (between low and high) the relation is not defined. Of course the high and low are not precise enough to use in a software, so we introduce two parameters, `low` and `high`, and we define the similarity of the i^{th} and j^{th} emails as follows

$$iRj \quad \begin{cases} \text{holds} & \text{if } x_{ij} > high \\ \text{undefined} & \text{if } low \leq x_{ij} \leq high \\ \text{doesn't hold} & \text{if } x_{ij} < low \end{cases}$$

With the election of these parameters we can control the relation. If we rise the values of parameters, then we narrow the relation. This reduces the number of errors of the first kind (when we treat letters similar while they dissimilar), but it increases the number of errors of the second kind (when we treat letters dissimilar while they similar). If we lower the parameters, then the changes are the opposite.

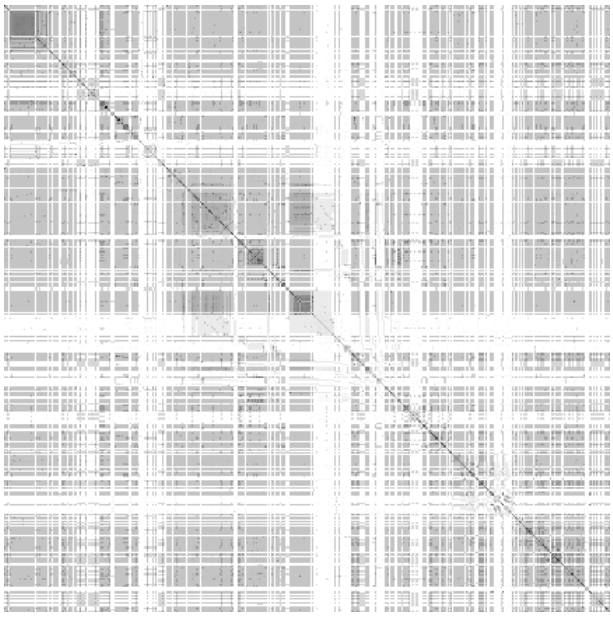

Figure 1: Similarity of tested emails

Some tests could help to set up the weights of the weighted sum and the two parameters.

5. Results

When we have the tolerance relation, the hard part is over. We have 433 emails, so we need to choose the best partition from all of them. About 10^{600} partitions exist, thus the exhaustive search is not suitable. The authors implemented several optimization methods [2], which gives near optimal results for combinatorial optimization problems.

Hence we used the rough clustering not on exact but on approximative solutions, but we are convinced, that the result of the rough clustering is near the optimal result. Table 1 shows our experiments. As the parameters became larger and larger, the size of the base sets became smaller and smaller, so we have more and more base sets.

6. Discussion

We have constructed a method to prepare objects for organization. How can we use the results? If we get a small or medium size base set, we can be sure that

low	high	biggest	no base sets
0.1	0.2	278	102
0.1	0.25	276	99
0.1	0.33	154	164
0.1	0.5	6	227
0.2	0.5	6	264

Table 1: Result of the rough clustering of emails

these objects are similar. If one of them got a tag, we can transfer it to the others, too. If we trust the clustering, this can be done automatically.

What can we do, if we have a strict clustering and many small base sets? In this case the object cannot inherit many tags from their base sets, so we need to check manually the objects of its upper approximation. Then the user decides whether to transfer some tags to the other objects or not.

As the tolerance relation may not be transitive, in some cases we are interested in the transitive closure of the upper approximations. For example, this could help to differentiate private and work mails.

By using a suitable mailer or picture-viewer, and altering the automated and manual steps, we can organize even a big collection within a short time. To explain this method in more detail we created a short video and put it on Youtube with the same title as this article.

7. Conclusion and further work

In this article we have shown a practical application of a fresh clustering method. This rough clustering is based on the partial version of the correlation clustering, and uses its equally good solutions to construct the lower and upper approximations. We used these approximations to transfer labels of emails to others to organize a whole collection.

The algorithm which constructs the tolerance relation uses two parameters. Our next task is to find the best combination of these parameters based on some restricted tolerance relation.

References

[1] Aszalós, L., and Mihálydeák, T. Rough clustering generated by correlation clustering. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. Springer, 2013, pp. 315–324.

[2] Bakó, M., and Aszalós, L. Combinatorial optimization methods for correlation clustering. In *Coping with complexity*, D. Dumitrescu, R. I. Lung, and L. Cremene, Eds. Casa Cartii de Stiinta, Cluj-Napoca, 2011, pp. 2–12.

[3] Bansal, N., Blum, A., and Chawla, S. Correlation clustering. *Machine Learning 56*, 1-3 (2004), 89–113.

[4] Csajbók, Z., Mihálydeák, T. Partial approximative set theory: A generalization of the rough set theory. In *International Journal of Computer Information System and Industrial Management Applications* **4** (2012) 437-444

[5] Csajbók, Z., Mihálydeák, T. A General Set Theoretic Approximation Framework. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager R. R. (eds): *Proceedings of IPMU 2012*, Catania, Italy, July 9-13, 2012, Part I, CCIS, Volume 297, Springer (2012) 604–612

[6] McIlroy, M. Development of a spelling list. *Communications, IEEE Transactions on 30*, 1 (1982), 91–99.

[7] Shneiderman, B., Byrd, D., and Croft, W. B. Sorting out searching: A user-interface framework for text searches. *Communications of the ACM 41*, 4 (1998), 95–98.

[8] Pawlak, Z., Skowron, A. Rudiments of rough sets. *Information Sciences* 177(1) (2007) 3–27

[9] Polkowski, L. Rough Sets: Mathematical Foundations. In *Advances in Soft Computing*, Physica-Verlag, Heidelberg, 2002.