Proceedings of the 10^{th} International Conference on Applied Informatics
Eger, Hungary, January 30–February 1, 2017. pp. 15–22
doi: 10.14794/ICAI.10.2017.15

Visualization of tolerance relations

László Aszalós, Dávid Nagy

University of Debrecen, Faculty of Informatics
aszalos.laszlo@inf.unideb.hu
nagy.david@inf.unideb.hu

Abstract

Previously we generated tolerance relations from the distance of objects. In this paper the situation is reversed: from any tolerance relation we generate a 2D representation in a such a way that the nodes of similar objects are close, while the nodes of different objects are far from each other. This placement is inspired by physics, and the location of the objects changes dynamically with the changes of the relation. As each undirected graph could be treated as a tolerance relation, results can be used widely.

1. Introduction

One of the main tools of data mining is cluster analysis. Here the set of objects are grouped in such a way, that the objects in the same group are more similar to each other than to those which are in other groups. Usually, the similarity and dissimilarity is based on the numbers describing the objects. But there are cases, where the objects cannot be described with numbers. Think of humans for example. It is hard to detail people by numbers, but we judge the similarity of persons, e.g. parents and children. Of course these opinions may vary, some can treat the father and its son as similar, while others treat them dissimilar.

If we want to formulate similarity and dissimilarity using mathematics, we need a tolerance relation. If this relation holds for two objects, we say that they are similar; and if this relation does not hold, we say that they are dissimilar. Of course, each object is similar to itself, so the relation is reflexive, and it is easy to show that it is symmetric too. However, we cannot go much further, e.g. the transitivity does not necessarily hold. If we take a human and a mouse, then due to their inner structures they are similar, this is reason why mice are used at drug experiments. Similarly, a human and a Paris doll are similar due to their shapes, this is the reason why dolls are used in show-windows. But there is no similarity between a mouse and a doll.

Clustering based on tolerance relation was introduced by Bansal (who gave a rough solution to it too) in [5] and named as *correlation clustering*, but Zahn drew

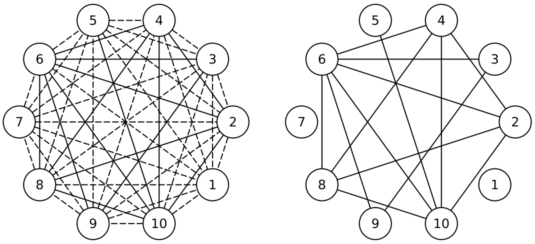

Figure 1: Tolerance relation based on GCD.

up this question from a mathematical point of view several years before in [7]. The authors gave several approximative solving methods for correlation clustering [2, 3, 4].

At presenting these solving methods, visualizing the tolerance relation proved problematic. We used ad hoc solutions, but it was only partially successful. It took too much human input, and it made it easy to make mistakes. Let us see a problem, that can be formulated easily: we treat two natural numbers similar, if their greatest common divisor is bigger than 1. And we treat them dissimilar, if their greatest common divisor is 1. The analysis of this problem and its surprising solution can be found in [1].

If we take this relation on numbers 1, 2, . . . , 10, we can construct the picture on the left in Fig. 1. Here the circles representing the numbers are positioned on a circle, to make it easy to connect the numbers. In this picture the similarity is denoted by solid lines, and the dissimilarity by dashed ones. This picture is transparent, but imagine a similar picture denoting the tolerance relation of numbers 1, 2, . . . , 100! Why not leave out the dashed lines? The picture on right in Fig. 1 shows only the similarities. It is slightly more understandable, but it is hard to see the nexus. If we reposition the numbers, everything becomes clearer. The numbers on the left on the circle have no similarities on the picture on the left of Fig. 2, and we can easily discover the groups 2-4-6-8-10, 3-6-9, 5-10. If we dispose of the circle, the structure could become even clearer. The software yEd produced the picture in Fig. 2 on the right.

Numbers are easy to compare. However it is not certain that 2 random objects are comparable. Or they are comparable, but nobody compared them yet. Hence the relation can be partial. This means that we have three cases: similar-relation holds, dissimilar-relation does not hold, unknown-relation undefined. We can visualize this with three colors, or three types of lines. If we do not draw the lines for unknown relations, then the solid and dashed lines are enough.

In the popular graph visualization methods (force directed graph) edges are modeled by springs, and the nodes are electrically charged particles. In these methods the similarity (the edge between two nodes) is handled with springs, and the dissimilarity (the absence of the edge between two nodes) is handled with electricity. The graphs visualized with these methods are sparse graphs, i.e. the number of edges is a linear function of the number of nodes.

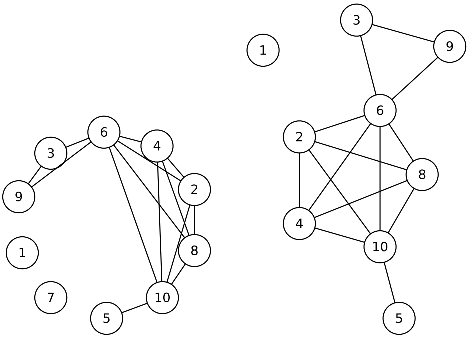

Figure 2: Repositioned tolerance relation of GCD

In our case we use three different kinds of springs according to the three different values of the partial tolerance relation. With these three values the graph of this relation is a dense one (all pairs of nodes are somehow connected), i.e. the the number of edges is a quadratic function of the number of nodes.

The structure of the article is the following: in the next section we present the physical background. In Section 3 we demonstrate our method by arranging the nodes of some specific relation. Finally, we conclude our results, and list our future research directions.

2. Metaphor from physics

We have used physical metaphors to solve the problem of correlation clustering, where similar objects attract each other and dissimilar objects repulse each other. We use the same here, but we do not group the objects, but arrange them on the plane (or in space). Our requirements are:

- similar objects get close, and
- dissimilar objects get far from each other.

Néda at al. presented a model using electric particles to solve the problem of correlation clustering in [6]. In this model, the particles could move on a circle based on the superposition of the forces acting on the particle.

As we wrote before, we use imaginary springs. Each node of the graph moves by the superposition of the forces of its springs. To simplify the problem in this section we use total tolerance relations, i.e. any pair of objects are comparable (similar or dissimilar). To get a suitable location for each of our objects, we have the following constraints:

- we do not like, if some object hide the others, so we fix an optimal distance (c) for similar objects.

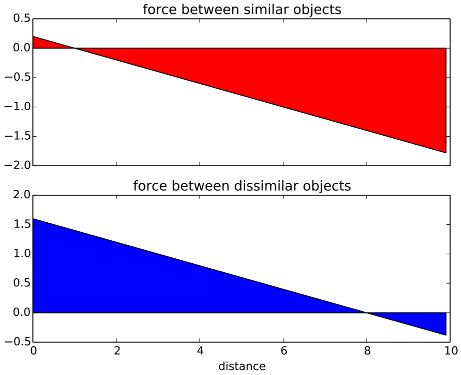

Figure 3: Spring functions

- to get a finite picture, we determine an optimal distance (C) for dissimilar objects.

We can translate this for springs: we have short springs for similar objects, and long springs for dissimilar objects. By Hooke's law, the force needed to extend or compress a spring by some distance d is proportional to that distance: $F = kd$.

Some placement of the objects can be treated as a network of these short and long springs. If two similar object (connected by a short spring) are closer than c, then they repulse, and if they are farther than c then they attract each other. For the long springs the same hold, but with C instead of c, as Fig. 3 shows. Therefore we introduce two functions: $f(\mathbf{d}) = (c - |\mathbf{d}|)$ and $f'(\mathbf{d}) = (C - |\mathbf{d}|)$, where $\mathbf{d}$ is the (distance) vector between two objects.

We use the periodic (sinusoidal) motion of a mass on a spring. We want to get a location of objects and not a motion, so we need some attenuation, a negative feedback. After some trial, we found that the cube of the distance is in some sense similar to the Coulomb's rule. Finally the resultant is the superposition of the forces:

$$\mathbf{F_i} = \sum_j \frac{f(\mathbf{d_{ij}})\mathbf{d_{ij}}}{|\mathbf{d_{ij}}|^3} + \sum_l \frac{f'(\mathbf{d_{il}})\mathbf{d_{il}}}{|\mathbf{d_{il}}|^3} \tag{2.1}$$

here the first part is summing for the objects similar, and the second part is for the objects dissimilar to object i. Our attenuation has a disadvantage, at small distances d_{ij} we may get very big numbers. But it has a great advantage: the system easily gets close to the equilibrium.

The authors in general work with partial relations, where it is not necessary to have a relation (or it is not known) between two objects. As we have no constraints on the unrelated objects, they can be at any distance from each other, so they can be positioned at the same place and partially or totally hide each other, or they can be very far from each other, so the picture can become very big. To solve these problems, we introduce a new spring function $\widehat{f}$ for the third type of springs (used for undefined values). This function is similar to the modified f', but the optimal distance is the interval $[c, C]$. If $d < c$ then $\widehat{f}(d) > 0$, and if $d > C$, then $\widehat{f}(d) < 0$, i.e. for small distances it repulses and for big distances it attracts.

3. Our results

After presenting the algorithm and its background, it is time to show it in practice. The algorithm can be downloaded from:
`https://github.com/aszalosl/visualize_tolerance`
We show the resulting image for some simple, but typical tolerance relation. Let us start with the *snake*, where adjacent objects are similar, and the others are dissimilar. We could think that the result is a straight line. Although the left side of Fig. 4 shows that this theory does not hold. If the dissimilar objects get too far from each other, then they attract, so we get an arch. In case of a non-defined relation for non-adjacent objects, we get a different image, because these objects do not repulse each other, so the snake can move in any directions, as right side of Fig. 4 shows.

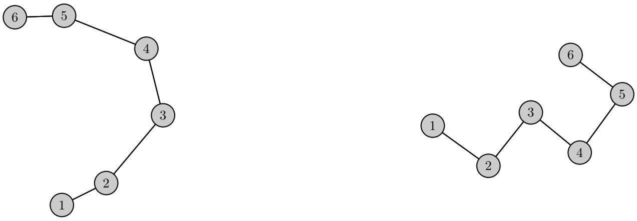

Figure 4: Total and partial snake.

Let us see some total trees! Here each non-leaf node has exactly three successors. Fig. 5 shows them. In the first two pictures the non-adjacent nodes are dissimilar, and in the last one the non-adjacent nodes are unrelated. The tree in the left is totally symmetric (if we omit the difference of the size of nodes according to one and two digits). But not every run gives such a nice picture. The middle one was generated with the same parameters as the left one, but the nodes 8 and 13 got to the wrong place at the beginning, and since this layout is stable, these nodes cannot escape. On the last picture the non-adjacent nodes, since unrelated, do not need to get far from the other nodes, e.g. nodes 3 and 4 comply the minimal distance constraint, so they are in stable state.

Of course the shapes of these pictures can be formed by changing the values of parameters c, C and C'. We tested several combinations, to get specious pictures, e.g. in case of Fig. 6. Here, we presented the GCD relation, but did not connect the similar nodes. From the numbers the reader can reconstruct these lines. The result of the correlation clustering for a few nodes gives a partition where each prime number (plus the 1) has a cluster, and each number gets into its smallest prime divisor's cluster. Although we do not like the collision of nodes (when they partly hide each other), but in this case it clarifies the picture.

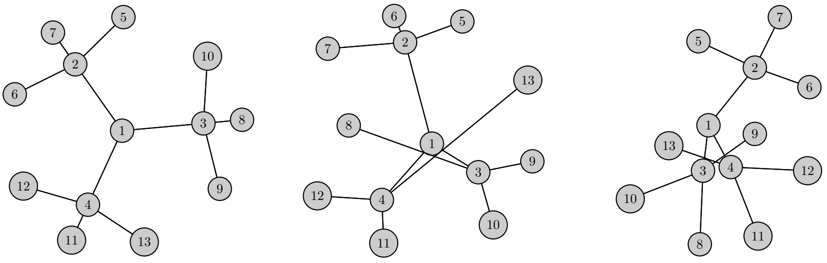

Figure 5: Total trees.

In the middle of the picture are the even numbers. On the right are the multiplies of three. 5 is just below 25 (we know it from the data of the image), as 7 is below 49. As the multiplicity of the divisors is not counted, the power of primes get to almost the same position (as 5 and 7 have shown). On the edge of the picture are the large prime numbers, because they differ from every other number, and as they dissimilar to each other, they are positioned uniformly. It is worth to examine the subtleties of the picture: 35 is positioned between the numbers of clusters of 5 and 7, but it is slightly moved towards the even numbers, because there are eight similar number (five numbers are divisible by five and three are divisible by seven), and three similar number in the cluster of 3. Similarly the number 50 is slightly moved towards the numbers of cluster of 5 and the number 48 towards the numbers of cluster of 3. By zooming the picture we could explore other subtleties, too.

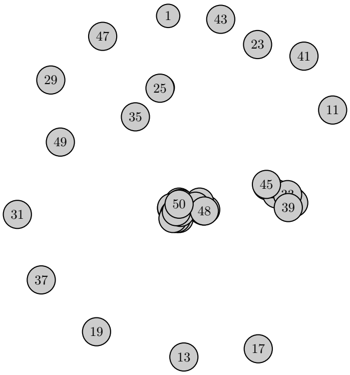

Figure 6: Visualization of the relation GCD.

Finally let us consider a picture which does not use abstract concepts of number theory, but comes from real life. Fig. 7 denotes the members of two departments. We treat two researchers as similar, if they are co-authors, and dissimilar if there is no such third person who is co-author to both. Of course this is a partial tolerance relation, because if A-B and B-C are pairwise similar, but A and C are not similar, then we cannot say that they are dissimilar according to the definition. Numbers

1-10 and 11-19 denote the members of the departments, respectively. The center of the picture is empty, hence the research areas are orthogonal. The numbers of the second department have higher densities, so their publication is stricter in the same themes mostly by the same co-authors. The numbers of the second department can be grouped into three clusters, and there are not many relations between these clusters.

Researchers 5, 6, 9 and 13 usually publish alone or with external colleagues, hence it is no wonder that they are alone in the picture. Researchers 1, 2, 7 and 8 wrote a common article, so they construct a strong core. As they publish with other authors too; they are positioned more widely on the picture according to the repulsion of other co-authors. From this group 1 and 2 are the only co-authors of 4. Moreover 4 is the regular co-author of 10 and has no other co-author. Hence 7 and 8 repulse 10, who gets far from 4, and the attraction of 10 moves 4 away from its other co-authors. In case of the other department, there is an attraction between 12 and 19 (they moved toward each other), but the chain of co-authors generates a repulsion, so they cannot get any closer.

This simple method could visualize complex systems, so we are planning a similar, faculty-wide image construction. In this model we did not take into account the quality and quantity of the common publications nor the date of these publications. Maybe this could tincture the image of the relations.

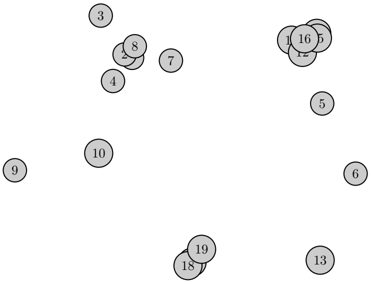

Figure 7: Research activities of two departments.

4. Conclusion, further research

In this article we gave a method which is able to represent a tolerance relation based on similarity and dissimilarity in 2D. The method is a variant of force-based method, which works on total and partial tolerance relations. We presented the algorithm, and applied it to some specific tolerance relations.

By changing the tolerance relation the method could change a given location, so the method could be used in an online system too. We presented the algorithm in 2D, but it can be easily extended into 3D. One run of Algorithm 1 takes n^2 steps, where n is the number of objects. We need to run this algorithm several hundred times, to get a suitable location. If n is big, this means a lot of calculations. We are sure that the tricks used at solving n-body problems, can be used here with a little adaptation. For example, similar and nearby objects (the members of a cluster) can be united and handled together by the addition of the forces. We have fast methods for clustering, so maybe this approach could speed up the calculations.

In the future we would like to examine, whether a higher dimension gives enough freedom to eliminate the local optimums, or just decreases their numbers, and by reducing the number of dimensions could we transform a solution in higher dimension into a spectacular result in 2D. Maybe this could help us in case of big datasets so that we could work with the data not in its original form, but in a prepared, partly interpreted form.

References

[1] Aszalós, L., Hajdu, L., Pethő, A.: On a correlational clustering of integers. Indagationes Mathematicae 27(1), 173–191 (2016)

[2] Aszalós, L., Kormos, J., Nagy, D.: Conjectures on phase transition at correlation clustering of random graphs. Annales Univ. Sci. Budapest., Sect. Comp (42), 37–54 (2014)

[3] Aszalós, L., Mihálydeák, T.: Correlation clustering by contraction. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015. pp. 425–434. IEEE (2015), http://dx.doi.org/10.15439/2015F137

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

[5] Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56(1-3), 89–113 (2004), http://dx.doi.org/10.1023/B:MACH.0000033116.57574.95

[6] Néda, Z., Sumi, R., Ercsey-Ravasz, M., Varga, M., Molnár, B., Cseh, G.: Correlation clustering on networks. Journal of Physics A: Mathematical and Theoretical 42(34), 345003 (2009)

[7] Zahn, Jr, C.: Approximating symmetric relations by equivalence relations. Journal of the Society for Industrial & Applied Mathematics 12(4), 840–847 (1964), http://dx.doi.org/10.1137/0112071